JPL D-7669, Part 2

Planetary Data System Standards Reference

October 8, 2012
Version 4.0.6

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

PDS4 build 3a, based on:

Information Model Specification
version 0.3.0.0.a
generated Wed Sep 19 10:02:45 PDT 2012

schema files
built from the PDS4 ontology model V0.3.0.0.a
generated Wed Sep 19 10:02:44 PDT 2012

Protegé ".pont" file
version 3.4.7
build 620
generated Fri Sep 14 18:01:07 PDT 2012

Contents

Change Log

Version	Section (old/new)	Change
4.0.1	Appendix A	Added examples for Image_Grayscale
4.0.2.	Appendix A	Temporarily dropped appendices
		(other changes not fully documented)
4.0.3	*throughout*	Changed references to external documentation to clickable links.
		Updated format of citations and made them clickable to bibliography
		Introduced global acronym handling and made acronyms clickable to definitions listed in appendices
		Made hidden hyperlinks explicit and therefore accessible in printed versions of the document, except where links were non-essential and provided solely for convenience of online readers
		Fixed text wrapping on long URLs that ran off page edge
	1.1/*	**PDS Data Policy** section removed.
	1.2/1.1	List of documents dropped; text updated to refer to complete list in bibliography.
	1.4/1.3	Deleted "standards" from third line.
	1.5/1.4	Completely re-written.
	1.6/*	All references in section 1.6 were transferred to the bibliography and the section was deleted.
	2 & 3	Switched the orders of chapters 2 and 3.
	2/3	Made numerous changes to opening paragraphs of chapter.

2.1/3.1	Significant changes throughout section, including dropping "errata.txt", changing name of "aareadme.xml" and changing cardinality values of files and subdirectories in root directory.
2.2/3.2	Changed "standard product label values" to "product label values"
2.5/3.5	Changed "standard data products" to "regular data products"
2.6/3.6	Changed "standard data products" to "basic data products"
2.7/3.7	Added required files table and completed text.
3.4/2.4	Changed "collections and standard products" to "collections, which in turn identify all of the basic products".
4	Changed "classes, elements, and standard values for the elements" to "classes, attributes, and standard values".
	Changed "classes, elements, and standard values" to "classes, attributes, and standard values."
	Significantly re-wrote introductory section.
	Updated label structure figure.
*/4.1	Added new section **The XML Declaration and Schema Reference**
4.1/4.2	Added text, label snippet, and table of attributes.
4.3/4.4	Added text and label snippet.
4.3.1/4.4.1	Added text and label snippet.
4.3.2/4.4.2	Added text and label snippet.
4.5.1.1/4.6.1.1	Deleted section **The Data_Location Class**.
*/4.6	Added new section **The Closing Tag**.
*/4.7	Added placeholder section **Local Data Dictionaries**.
5	Added "base classes" to end of last sentence.
5.4	Changed two instances of "encoded stream base" to "encoded byte stream".
6.1	Changed title from **Attribute Data Types** to **Character Data Types**.
	Modified text to include character tables.
	Re-organized data_types into subsections.
	Deleted ASCII_Date_Time_UTC.
	Changed ASCII_Integer_Base* to ASCII_Numeric_Base*.
	Added ASCII_Mask.

	Updated max value of ASCII_NonNegative_Integer from 2147483647 to 4294967295.
6.2/*	Deleted section **Character Data Types**.
6.3/6.2	Added text.
6.3.1.1/6.2.1.1	Changed “Least Significant Bit (LSB)” to “Least Significant Byte (LSB) first (also known as *little-endian*)”
	Dropped the SignedByte data type.
6.3.1.2/6.2.1.2	Changed “Least Significant Byte first (LSB) format” to “LSB format”
6.3.1.3/6.2.1.3	Changed “Most Significant Byte first (MSB)” to “Most Significant Byte (MSB) first (also known as *big-endian*)”
	Dropped the SignedByte data type.
6.3.1.4/6.2.1.4	Changed “Most Significant first (MSB) format” to “MSB format”
6.3.3/6.2.3	Added figures for complex types.
*/6.2.4	Added subsection heading for Boolean binary type.
7.4	Sixth bullet under **Rules**, changed “alphabetical” to “alphanumeric”.
8/*	**Astronomical Nomenclature** chapter removed.
10.1/9.1	Last sentence: changed “late 2008” to “early 2011”
10.1.1/9.1.1	Second paragraph: updated WGCCRE list to include 2011 report
10.2/9.2	Second line: changed “bodys” to “body’s”
	Second paragraph: fixed broken Greek symbols
	Third paragraph, last sentence: changed “See the Planetary Science Data Dictionary (PDS, 2008), chapter 2” to “See chapter 12 of this document”
	Fourth paragraph, first sentence: fixed incorrect citation; changed “(IAU, 2000)” to “(IAU, 2002)”
	Fourth paragraph: changed “Earths” to “Earth’s”
10.3/9.3	First paragraph: changed “Suns” to “Sun’s” (twice)
10.4.1/9.4.1	Throughout: fixed missing degree symbols
	First paragraph: changed “bodys” to “body’s”
	Second paragraph: added “the” before “Sun”
	Second paragraph: fixed Greek symbol

		Third paragraph: changed “existing PDS data sets” to “existing sets of data archived with the PDS”
	10.7/9.7	First paragraph: fixed superscripted text
	13/12	Added note about SPICE time systems after final paragraph.
	13.3.1/12.3.1	Changed case of attributes from upper to lower.
		Changed “elements” to “attributes”.
		Removed “and catalog files”.
	13.3.2/12.3.2	Changed “elements” to “attributes”.
	13.3.2.1/12.3.2.1	Changed case of attributes from upper to lower.
		Changed “elements” to “attributes”.
	13.3.2.4.1/12.3.2.4.1	Replaced all instances of “areocentric” with “planetocentric’.
		Replaced all instances of “ARA” with “RA”
		Changed case of attributes from upper to lower.
		Changed “element” to “attribute”.
		Changed “Zaxis” to “Z-axis”
		Added line breaks and indenting in LTST equations.
	13.3.2.4.2/12.3.2.4.2	Replaced “areocentric” with “planetocentric”
	appendices	Fixed chapter and section numbering in appendices
	Appendix A	Re-attached Appendix A, “Digital Object Classes”
		Completely re-wrote introductory section.
		Fixed automation code to include only terminal user classes
		Fixed automation code to include inherited attributes and associations
		In “Attributes & Associations” sections, changed redundant “Req?” column to “Values” column
		Dropped schema samples
		Improved automated generation of XML examples
	Acronyms	Added list of acronyms
	Bibliography	Added bibliography

Version	**Section (old/new)**	**Change**

4.0.4	*preface*	Removed “Responses to Build 1B Assessment”
		Removed “Responses to Build 1C Assessment”
	2/*	Removed Chapter 2, “Data Objects and Products”
		(other changes not fully documented)

Version	**Section (old/new)**	**Change**
4.0.5	*all*	numerous
	1.5	Changed “International Standardization Organization” to “International Organization for Standardization”

Version	**Section (old/new)**	**Change**
4.0.6	*all*	Updated all schema references to current version of schema.
	1.5	Added RFC 1321.
	1.6	Changed “schemata” to “schemas”.
	2	Collapsed each set of three inventory tables into a single table. Changed cardinality on single table to 1.
	2.1	Changed “schema” to “xml_schema”.
	Fig. 2.2	Removed “file_specification_name” from every instance of “Bundle_Member_Entry”.
		Removed “external_standard_version_id” from “Stream_Text” class and updated value of “external_standard_id” from “HTML” to “HTML 3.2”.
		Added “record_delimiter” to “Stream_Text”.
		Changed name of schema collection to xml_schema.
		Re-ordered xml_schema and spice collections.
		Changed member_status of spice collection from “Secondary_No_File” to “Secondary”.
		Added “Bundle_Member_Entry” for geometry collection.
	2.5	Added context product files for agency, facility, pds affiliate, pds guest, and telescope.
		Re-ordered node and other context product files.

Fig. 2.3	Modified sample inventory table to remove file specification names and add member status.
Fig. 2.4	Modified sample inventory table to remove file specification names and add member status.
	Changed constituent files in xml_schema collection.
	Updated caption consistent with above updates.
Fig. 2.8	Changed from schema collection label to data collection label.
	Added unit="byte" to file_size attribute of File class.
	Changed local_identifier for File class from "PRIMARY MEMBER INVENTORY FILE" to "INVENTORY TABLE FILE".
	Changed local_identifier for Inventory class from "PRIMARY MEMBER INVENTORY" to "INVENTORY TABLE".
	Added external_standard_id attribute to the Inventory class.
	Re-ordered records and encoding_type attributes in the Inventory class.
	Updated number of records from 2 to 16661 in Inventory class.
	Added field_delimiter attribute to Inventory class.
	Changed Record_Character class in Inventory to Record_Delimited.
	Changed Field_Character class in Inventory to Field_Delimited.
	Updated fields in Inventory to Member_Status and LIDVID_LID.
	Updated reference_type value of Inventory from "inventory_has_LIDVID_Primary" to "inventory_has_member_product".
	Deleted second File_Area_Inventory from Product_Collection.
2.11	Deleted separate Schematron label file, "PDS4_PDS_0300a_sch_label.xml", from xml_schema directory listing.
Tab. 3.1	Changed "Node Area" to "Discipline Area".
	Changed "Observation Area" to "Context Area / Observation Area". Changed check marks to "C"s or "O"s.
	Added Context_Area to Product_Bundle.
	Added Context_Area to Product_Collection.
	Added Context_Area to Product_Document.
	Dropped "Subject Area".
5.1.2	Removed footnote from ASCII_Date.

	Removed old footnote from ASCII_Date_Time.
	Added new footnote about truncation of date/time values to ASCII_Date_Time, ASCII_Date_Time_DOY, and ASCII_Date_Time_TMD.
	Removed footnote from ASCII_Date_Time_UTC. Modified definition to explain reason for inclusion of this format.
5.1.4	Added ASCII_LIDVID_LID data format.
5.3.1.1	Added SignedByte integers.
5.3.1.2	Added placeholder for UnsignedLSB8 integers.
5.3.1.3	Added SignedByte integers.
5.3.1.4	Added placeholder for UnsignedMSB8 integers.
5.3.2	Added note about needing to update for MSB and LSB real types.
5.3.3	Added note about needing to update for MSB and LSB complex types.
5.3.4	Deleted Boolean data type.
	Added placeholder for Bit Strings, including both signed and unsigned.
5.4/5.1.5	Moved “Multi-Valued Data Types” section into section on “Attribute Value Types”.
5.4.1/5.1.5.1	Added text plus table showing vector data types. Also added footnotes explaining which are obsolete and which are likely to be used.
5.4.2/5.1.5.2	Added note that quaternions not yet implemented.
6.1	In fifth paragraph, changed “discipline nodes” to “disciplines”.
	Added “calib”, “cart”, and “geom” namespaces to Table 6.1.
	Combined the pds_affiliate and pds_guest collections into a single “personnel” collection.
	Added “agency” and “other” collections.

II/III	Data Content Standards part of document has been broken into two parts: "Data Content Standards" and "Discipline Standards". Chapters 7 (Calibration) and 8 (Cartographic Standards) have been moved to Discipline Standards and re-numbered as chapters 11 and 12, respectively. Chapter 13 (Geometry) has been added. Chapter 7 is now Context Documentation (renamed from old 9, Content Information). Chapter 8 is now Other Documentation (renamed from old 10, Documentation). The Time Standards chapter has been renumbered from 11 to 9, and the Units of Measurement chapter has been renumbered from 12 to 10.
9/7	Significant re-write of entire chapter. The available list of context products has been increased, and textual descriptions have been added to each. The introduction to context products has also been fleshed out.
10/8	Removed "Volume information" from the list of typical archive documents.
	Re-wrote fourth paragraph to make it consistent with PDS4 archiving of document products.
10.1/8.1	Deleted section entitled "Data Set Description".
10.2/8.2	Deleted section entitled "Instrument Description".
10.3/8.3	Deleted section entitled "Instrument Host Description".
10.4/8.4	Deleted section entitled "Investigation / Mission Description".
Acronyms	Added "ESA".

Chapter 1

Introduction

The Planetary Data System (PDS) Standards Reference is a complete specification for version 4 of the PDS standards. These standards are used to design data storage formats and encode descriptive labels for data stored in the PDS.

Note that version 4 of the PDS standards is *not* backwards compatible with version 3. All version 4 data can be described using version 3 labels, but the converse is not true.

This document *does not* provide a formal definition of the grammar of the eXtensible Markup Language (XML), which is used to encode the PDS4 standards; guidance on this aspect of PDS labels is provided in the Data Provider's Handbook (DPH).

1.1 Purpose

This document is intended to serve as the reference document detailing PDS standards used in the preparation of PDS compliant data. It is to be used within the context of the PDS4 document suite, described in the *Introduction to the PDS4 Document Set*. (Please see the bibliography at the end of this document for a full listing of PDS documents available.)

1.2 Scope

The information included here constitutes Version 4.0 of the Planetary Data System data preparation standards for producing archive quality data sets. This document covers the conceptual composition of an archive, its physical layout, and the current technology standards used for implementing the data and meta-data.

1.3 Audience

This document is intended primarily to serve the community of scientists and engineers responsible for preparing planetary science data sets for submission to the PDS. These include restored data from the era prior to PDS or from earlier versions of the PDS, mission data from active and future planetary missions, and data from earth-based sites. The audience includes personnel at PDS discipline and data nodes, mission principal investigators, and ground data system engineers. This document is intended for use by those people already somewhat familiar with the process of archiving data with the PDS. (Those new to the PDS should first read the DPH.)

1.4 Document Organization

This document is divided into three main sections. The first of these, “Archive Structure Standards”, provides detailed information about the structure of all of the components of a PDS archive: labels, data, bundles, collections, and overall organization. The second major section, “Data Content Standards”, is more focused on the nature of *what* is stored in PDS archives. This includes guidance on necessary calibration information, contextual information, and documentation, as well as how to populate cartographic information, date and time values, and various nomenclature rules. Finally, the Appendices contain detailed listings of specific classes used in describing PDS data.

1.5 External Standards

External standards which apply to the content of this document:

American National Standards Institute (ANSI):

- ANSI INCITS 4-1986 (R2007) *Information Systems - Coded Character Sets - 7-Bit American National Standard Code for Information Interchange (7-Bit ASCII)*

Consultative Committee for Space Data Systems (CCSDS):

- CCSDS 641.0-B-2 *Parameter Value Language Specification (CCSD0006 and CCSD0008)* (also available as ISO 14961:2002)

Institute of Electrical and Electronics Engineers (IEEE):

- IEEE 754-2008 *Standard for Binary Floating-Point Arithmetic*

International Organization for Standardization (ISO):

- ISO 646:1991 *ISO 7-bit coded character set for information interchange*
- ISO 8601:2004 *Data Element and Interchange Formats – Representations of Dates and Times*
- International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 10646:2012 *Information technology – Universal Coded Character Set (UCS)*
- ISO/IEC 11179-3:2003 *Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes*
- ISO/IEC 11404:2007 *General-Purpose Datatypes (GPD)*
- ISO 14721:2003 *Open archival information system – Reference model*
- International Standards Organization / Technical Standard (ISO/TS) 15000-3:2004 *electronic business eXtensible Markup Language (ebXML) – Part 3: Registry information model specification (ebRIM)*
- ISO/TS 15000-4:2004 *ebXML – Part 4: Registry services specification (ebRS)*

Internet Engineering Task Force (IETF)

- Request for Comments (RFC) 1321 *The MD5 Message-Digest Algorithm*, April 1992

National Institute of Standards and Technology (NIST):

- NIST Special Publication 330 *The International System of Units (SI)*, United States version of the English text of the eighth edition (2006), Issued March 2008

World Wide Web Consortium (W3C):

- *XML 1.1* 2nd ed., August 16, 2006
- *XML Schema Part 0: Primer* 2nd ed., October 28, 2004
- *XML Schema Part 1: Structures* 2nd ed., October 28, 2004
- *XML Schema Part 2: Datatypes* 2nd ed., October 28, 2004

1.6 Online Document Availability

All PDS documents pertaining to archive preparation are available online. Information on accessing these references may be found on the PDS website at the following URL:

http://pds.nasa.gov

To obtain a copy of these documents or for questions concerning these documents, contact the PDS Operator (at pds_operator@jpl.nasa.gov, 818-393-7165) or any PDS data engineer.

The examples provided throughout this document are based on both existing and planned PDS archive products, modified to reflect the current version of the PDS Standards. Discipline-specific extensions to the high-level classes defined in this document are created and augmented from time to time, as user community need arises. To check the current status of any discipline- or mission-specific class definition, consult a PDS data engineer.

Additional schemas and examples are available online at the following URL:

http://pds.nasa.gov/pds4/schema/released/

Part I

Archive Structure Standards

Chapter 2

Archive Organization and Directory Contents

An archive *bundle* is the primary construct for delivering digital data to the PDS.

Bundles have a simple hierarchical structure. A bundle has one or more member *collections*, each of which has one or more member *basic products*.

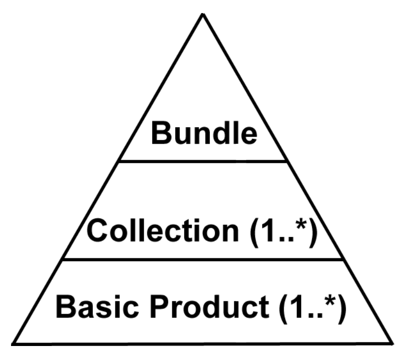


Figure 2.1: Overall bundle structure.

Data is organized into collections based on the type and function of the data. So, for example, observational data from a particular instrument and of a particular processing level might constitute one collection; low resolution representations of the observational data for browsing purposes might constitute a second collection; documentation describing the data collection and calibration process might constitute a third collection; geometry information related to the observational data represented in SPICE format might constitute a fourth, etc.

Both collections and basic products may be either *primary* or *secondary* members of their respective bundles or collections. A primary product is one which is being delivered to the PDS for the first time. A secondary product is one which has already been delivered to and registered with the PDS, but which is now being associated with a new bundle or collection.

Note that data providers are not obligated to include physical copies of secondary member products in their archive bundles, since they are already present within the PDS repository as primary members of another bundle. However, for the sake of the convenience of data users, it is permissible to include copies of these products. (PDS nodes, on the other hand, are obligated to deliver copies of these products when bundles are submitted to the deep archive.)

Data providers will typically deliver data to the PDS in a directory structure that matches the organization of the bundle. The data will be delivered to the deep archive in this format; however, PDS discipline nodes may or may not choose to retain this structure in their online repositories.

The following sections describe the contents of an archive repository starting with the root directory, followed by the contents of the subdirectories in alphabetical order. In the tables that follow, the following conventions have been observed:

- A **bold-faced** font is used for directory names, while plain type is used for file names. Fixed or mandated file names are shown in regular font; sample file names are shown in *italics*.

- Many of the file names contain version numbers of either the "x" or "x.y" variety. In all cases, even though a "1" or "1.0" is shown in the fixed file names, these numbers can of course be incremented in actual delivered archives. (For more information about versioning within PDS4, see section 6.3.1.)

- Where multiple file extensions are shown in square brackets, [ ], this indicates that either of the file extensions may be used, as appropriate to the format of the file. (For more details on reserved file extensions, see Table 6.2.)

- The "Class" identifies the PDS4 class used in the specified label to describe associated objects. Files for which no class is identified are described in separate label files.

- "Cardinality" refers to the number of instances of a particular file or directory type that may occur in an archive. Where a single digit appears, the archive must include precisely that number of files (or directories). Where a range of numbers is specified (using the notation m..n), then any number of files in that range may be included. A minimum of "0" means that the file or directory is optional; a maximum of "*" means that there is no limit on the number of files that may be included.

- "Name Fixed?" provides an indication of whether or not the associated directory or file name is mandated ("fixed") or up to the discretion of the data provider ("not fixed"). In a number of cases, part of the file name is mandated and part is up to the discretion of the data provider (indicated by "prefix fixed"). This typically occurs in instances where more than one of a particular type of directory or file may be included in an archive. The mandated part of the file name indicates the type of file (or directory) and the discretionary part of the file name is used to distinguish between files (or directories) of similar type.

2.1 Root Directory

The root directory corresponds to the top level of an archive bundle. It contains the bundle product and directories corresponding to each of the member collections.

File or Directory Name	Class	Cardinality	Name Fixed?
root	–	1	not fixed
bundle_1.xml	Product_Bundle	1[1]	fixed
readme_1.0.[html,txt]	–	0..1	fixed
browse[_✶]	–	0..✶	prefix fixed
calibration[_✶]	–	0..✶	prefix fixed
context[_✶]	–	0..✶	prefix fixed
data[_✶]	–	0..✶	prefix fixed
document[_✶]	–	0..✶	prefix fixed
geometry[_✶]	–	0..✶	prefix fixed
miscellaneous[_✶]	–	0..✶	prefix fixed
spice_kernels[_✶]	–	0..✶	prefix fixed
xml_schema	–	1	fixed

[1]Note that while there may only be one *logical* bundle in a delivered archive, multiple versions of the bundle label may be present. Thus, it is permissible to have both “bundle_1.xml” and “bundle_2.xml” files in the root directory. The same is true for all other versioned files in an archive.

The bundle product is described by a label with the name “bundle_1.xml” (where “1” is the version identifier and may be incremented). This label provides the identifying information for the bundle and contains a list, in XML format, identifying all of the component collections of which the bundle is comprised.

The bundle product shall be implemented using the Product_Bundle class. There must be one, and only one, of these files included in each bundle (although see footnote to above table). The filename, ”bundle_1.xml”, is fixed.

A second, optional “readme” file provides a general overview of the bundle contents and organization in human readable format. It may also contain general instructions for use of the archive and contact information for data preparer or discipline node personnel. This file may be formatted either as plain text or HTML. Note that this file is an optional component of the bundle product, and is not considered to be a separate product. Thus, it shares a label with the bundle. It is described using the File_Area_Text class within the Product_Bundle. (See Figure 2.2.)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
    href="http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>

<Product_Bundle
    xmlns="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:pds="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://pds.nasa.gov/pds4/pds/v03
        http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.xsd">

  <Identification_Area>
    <logical_identifier>urn:nasa:pds:mars_mpf_imp_raw</logical_identifier>
    <version_id>1.0</version_id>
    <title>MPF Imager for Mars Pathfinder Raw Data Archive</title>
    <information_model_version>0.3.0.0.a</information_model_version>
    <product_class>Product_Bundle</product_class>
  </Identification_Area>

  <Bundle>
    <description>
      This archive contains the raw data recorded by the Imager for Mars Pathfinder
      mounted on the Mars Pathfinder Lander.  The data was submitted to the
      Planetary Data System (PDS) by Peter Smith, IMP Principal Investigator.  This
      archive covers the period from 1996-12-16 to 1997-09-27.
    </description>
    <bundle_type>Archive</bundle_type>
  </Bundle>

  <File_Area_Text>
    <File>
      <file_name>readme_1.0.html</file_name>
      <local_identifier>README FILE</local_identifier>
      <creation_date_time>2012-05-30T22:02:43Z</creation_date_time>
      <md5_checksum>fbba2b0a52d16f00bf4bd6346cb4ef53</md5_checksum>
    </File>
    <Stream_Text>
      <offset unit="byte">0</offset>
      <external_standard_id>HTML 3.2</external_standard_id>
      <encoding_type>Character</encoding_type>
      <record_delimiter>carriage_return line_feed</record_delimiter>
    </Stream_Text>
  </File_Area_Text>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:browse</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_browse_collection</reference_type>
  </Bundle_Member_Entry>
```

```
  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:context</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_context_collection</reference_type>
  </Bundle_Member_Entry>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:data</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_data_collection</reference_type>
  </Bundle_Member_Entry>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:document</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_document_collection</reference_type>
  </Bundle_Member_Entry>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:geometry</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_geometry_collection</reference_type>
  </Bundle_Member_Entry>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mpf:spice</lid_reference>
    <member_status>Secondary</member_status>
    <reference_type>bundle_has_spice_kernel_collection</reference_type>
  </Bundle_Member_Entry>

  <Bundle_Member_Entry>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:xml_schema</lid_reference>
    <member_status>Primary</member_status>
    <reference_type>bundle_has_schema_collection</reference_type>
  </Bundle_Member_Entry>
</Product_Bundle>
```

Figure 2.2: Example of an Bundle Label

2.2 Subdirectories

Beneath the root directory, each top level subdirectory has a one-to-one correspondence with the bundle's collections. Within each collection directory, the collection product is identified and described by an XML label with the name "collection_1.0.xml" (where "1.0" is the version identifier and may be incremented). The members of the collection are listed in one to three *inventory* tables.

The members of a collection are basic products. They may be primary, secondary, or a mixture of both. Basic products are primary members of the collection within which they first enter PDS4. Every basic product must be a primary member of one (and only one) collection. (Note that each new version of a basic product is considered to be a primary member of its collection.)

A basic product may be a secondary member of any number of collections. A collection which lists references to basic products already registered in PDS would identify those products as its secondary members. (Secondary member products might be included in a collection for a new purpose or because of their relevance to other collections in a bundle. For example, a collection of secondary products might consist of previously archived Voyager images of Saturn which have been identified as containing Saturn's rings within the field of view. Alternatively, a collection of primary products containing geometrically corrected images might be accompanied by a collection of secondary copies of SPICE files already archived by the Navigation and Ancillary Information Facility.)

Inventory tables identify the basic products contained in a collection. Primary member products are identified by a LIDVID. Secondary member products may be identified by either a LID or a LIDVID. (See section 6.2.1 for more information on LIDs and LIDVIDs.) Each collection shall contain one inventory table listing all of the primary members of the collection (named "collection_primary_1.0.tab"), one inventory table listing all of the secondary members identified by LIDVID (named "collection_secondary_lidvid_1.0.tab"), and one inventory table listing all the secondary members identified by LID (named "collection_secondary_lid_1.0.tab"). If basic products of a particular type are not included in the collection, then the inventory table for that type of member product shall not be included either.

Inventory tables are a subclass of the Table_Character class and therefore have records of fixed length. They contain one record for each basic product they identify. Each record in an inventory table which identifies primary members shall consist of the LIDVID for that member product, followed by the file specification name, in UNIX format relative to the directory containing the collection label, of the XML label for the basic product. The records of inventory tables identifying secondary members shall consist of only the LID or LIDVID. Examples of primary and secondary inventory tables are shown in Figures 2.3 and 2.4.

The label for each collection and its associated collection inventory tables shall reside in a top level subdirectory. The member products for each of the collections shall reside in lower level directories, named and organized according to the preference of the data provider. (An exception is made to this rule for collections containing less than roughly two dozen products, in which case the products and their labels may reside alongside the collection label in the top level subdirectory. See Figures 2.5 and 2.6.)

In cases where a bundle contains multiple collections of the same type, the names of the subdirectories containing these collections shall be distinguished by a suffix of the data provider's choosing appended to the mandated directory name. For example, if two calibration collections are present in the archive, the two calibration directories might be named **calibration_ground** and **calibration_flight**. (See Fig. 2.7.)

A sample collection label is shown in Figure 2.7.

2.3 Browse Directory

Each browse directory contains a browse collection. Browse collections contain "quick-look" products designed to facilitate use of the archive. Browse products and the browse directory are optional.

File or Directory Name	Class	Cardinality	Name Fixed?
browse[_✶]	–	0..✶	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
cruise	–	1..✶	not fixed
cruise_19961216193124_1.0.xml	Product_Browse	1..✶	not fixed
cruise_19961216193124_1.0.jpg	–	1..✶	not fixed
⋮	⋮	⋮	⋮

For each browse collection present in the archive, there shall be one **browse** directory, one collection label, and one to three inventory tables. The browse collection shall be implemented using the Product_Collection class.

If appropriate, each browse directory may contain multiple levels of sub-directories. The structure of these sub-directories is at the discretion of the data provider. A structure that parallels the structure of the data subdirectories is frequently utilized.

Browse products shall be implemented using the Product_Browse class.

```
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455934-DARK_CURRENT-0000000008<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455934-DARK_STRIP-0000000008<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455934-NULL_STRIP-0000000008<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455934-DARK_CURRENT-0000000009<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455940-DARK_CURRENT-0000000010<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455940-DARK_STRIP-0000000010<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455940-NULL_STRIP-0000000010<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455940-DARK_CURRENT-0000000011<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455947-DARK_CURRENT-0000000012<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1229455947-DARK_STRIP-0000000012<CR><LF>
```

Figure 2.3: Example of an inventory table identifying primary members of a collection

```
s,urn:nasa:pds:system_bundle:xml_schema:pds-xml_schema<CR><LF>
s,urn:nasa:pds:system_bundle:xml_schema:geom-xml_schema<CR><LF>
s,urn:nasa:pds:system_bundle:xml_schema:img-xml_schema<CR><LF>
p,urn:nasa:pds:mars_mpf_imp_raw:xml_schema:mpf-xml_schema<CR><LF>
```

Figure 2.4: Example of an inventory table identifying primary and secondary members of a collection

```
root
  - bundle_1.xml
  - readme_1.0.txt
  - browse
        - collection_1.0.xml
        - collection_1.0.tab
        - flight
              - flight_0001_1.0.jpg
              - flight_0001_1.0.xml
              - flight_0002_1.0.jpg
              - flight_0002_1.0.xml
                        ⋮
```

Figure 2.5: Subdirectory containing one collection with many basic products.

```
root
        ⋮
  - schema
        - collection_1.0.xml
        - collection_1.0.tab
        - imaging_dictionary_1.0.xsd
        - mpf_dictionary_1.0.xsd
        - mpf_imp_raw_1.0.sch
        - PDS4_PDS_0800k.sch
        - PDS4_PDS_0800k.xsd
```

Figure 2.6: Subdirectory containing one collection with few basic products.

```
root
  |       ⋮
  | - calibration_flight
  |         - collection_1.0.xml
  |         - collection_1.0.tab
  |         - cruise
  |                 - flightcal_product01.jpg
  |                 - flightcal_product01.xml
  |                 - flightcal_product02.jpg
  |                 - flightcal_product02.xml
  |                              ⋮
  |         - orbit_0001-0100
  |
  |
  | - calibration_ground
  |         - collection_1.0.xml
  |         - collection_1.0.tab
  |         - 2010-03
  |                 - 2010-03-14_0001.jpg
  |                 - 2010-03-14_0001.xml
  |                 - 2010-03-14_0002.jpg
                    - 2010-03-14_0002.xml
                                 ⋮
```

Figure 2.7: Two subdirectories containing collections of the same type.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
    href="http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>

<Product_Collection
    xmlns="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:pds="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://pds.nasa.gov/pds4/pds/v03
        http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.xsd">

  <Identification_Area>
    <logical_identifier>urn:nasa:pds:mars_mpf_imp_raw:data</logical_identifier>
    <version_id>1.0</version_id>
    <title>MPF Imager for Mars Pathfinder Data Files</title>
    <information_model_version>0.3.0.0.a</information_model_version>
    <product_class>Product_Collection</product_class>
  </Identification_Area>

  <Context_Area>
    <Time_Coordinates>
      <start_date_time>1996-12-16T19:31:24.600Z</start_date_time>
      <stop_date_time>1997-09-27T10:14:35.340Z</stop_date_time>
    </Time_Coordinates>

    <Primary_Result_Summary>
      <type>Image</type>
      <purpose>Science</purpose>
      <data_regime>Visible</data_regime>
      <processing_level_id>Raw</processing_level_id>
    </Primary_Result_Summary>

    <Investigation_Area>
      <name>MARS PATHFINDER</name>
      <type>Mission</type>
      <Internal_Reference>
        <lid_reference>urn:nasa:pds:context:mission:MARS_PATHFINDER</lid_reference>
        <reference_type>data_to_investigation</reference_type>
      </Internal_Reference>
    </Investigation_Area>

    <Observing_System>
      <Observing_System_Component>
        <name>MARS PATHFINDER LANDER</name>
        <observing_system_component_type>Spacecraft</observing_system_component_type>
        <Internal_Reference>
          <lid_reference>urn:nasa:pds:instrument_host.MPFL</lid_reference>
          <reference_type>is_instrument_host</reference_type>
        </Internal_Reference>
      </Observing_System_Component>
```

```
    <Observing_System_Component>
      <name>IMAGER FOR MARS PATHFINDER</name>
      <observing_system_component_type>Instrument</observing_system_component_type>
      <Internal_Reference>
        <lid_reference>urn:nasa:pds:instrument.IMP__MPFL</lid_reference>
        <reference_type>is_instrument</reference_type>
      </Internal_Reference>
    </Observing_System_Component>
  </Observing_System>
</Context_Area>

<Reference_List>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:geometry</lid_reference>
    <reference_type>collection_to_geometry</reference_type>
  </Internal_Reference>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:mpf:spice</lid_reference>
    <reference_type>collection_to_spice_kernel</reference_type>
  </Internal_Reference>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:mars_mpf_imp_raw:xml_schema</lid_reference>
    <reference_type>collection_to_schema</reference_type>
  </Internal_Reference>
</Reference_List>

<Collection>
  <collection_type>Data</collection_type>
</Collection>

<File_Area_Inventory>
  <File>
    <file_name>collection_1.0.tab</file_name>
    <local_identifier>INVENTORY TABLE FILE</local_identifier>
    <creation_date_time>2012-06-03T15:56:39Z</creation_date_time>
    <file_size unit="byte">160</file_size>
    <md5_checksum>50f0d0027fc47426923ffe2a37df01c5</md5_checksum>
  </File>
  <Inventory>
    <local_identifier>INVENTORY TABLE</local_identifier>
    <offset unit="byte">0</offset>
    <external_standard_id>PDS_DSV V1.0</external_standard_id>
    <encoding_type>Character</encoding_type>
    <records>2</records>
    <record_delimiter>carriage_return line_feed</record_delimiter>
    <field_delimiter>comma</field_delimiter>
    <Record_Delimited>
      <fields>2</fields>
      <maximum_record_length unit="byte">81</maximum_record_length>
      <Field_Delimited>
        <name>Member_Status</name>
        <field_number>1</field_number>
        <data_type>ASCII_String</data_type>
        <maximum_field_length unit="byte">1</maximum_field_length>
      </Field_Delimited>
```

```
        <Field_Delimited>
          <name>LIDVID_LID</name>
          <field_number>2</field_number>
          <data_type>ASCII_LIDVID_LID</data_type>
          <maximum_field_length unit="byte">77</maximum_field_length>
        </Field_Delimited>
      </Record_Delimited>
      <reference_type>inventory_has_LIDVID_Primary</reference_type>
    </Inventory>
  </File_Area_Inventory>
</Product_Collection>
```

Figure 2.7: Example of a Collection Label

2.4 Calibration Directory

Each calibration directory contains a calibration collection. Calibration collections contain calibration data and files necessary for the calibration of science data products. Calibration products and the calibration directory are required in archives where they are necessary for the proper understanding and interpretation of science data.

File or Directory Name	Class	Cardinality	Name Fixed?
calibration[_✶]	–	0..✶	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
preflight_200903	–	1..✶	not fixed
calibration_20090314a.xml	TBD	1..✶	not fixed
calibration_20090314a.tab	–	1..✶	not fixed
⋮	⋮	⋮	⋮

For each calibration collection present in the archive, there shall be one **calibration** directory, one collection label, and one to three inventory tables. The calibration collection shall be implemented using the Product_Collection class.

If appropriate, each calibration directory may contain multiple levels of sub-directories. The structure of these sub-directories is at the discretion of the data provider.

2.5 Context Directory

Each context directory contains a context collection. Context collections consist of XML labels for various physical and conceptual objects identified within the PDS4 registry. (This includes physical objects such as instruments, spacecraft, and planets, and conceptual objects such as missions and PDS nodes.) These context products provide only the information necessary to identify and classify these objects; they are not intended to include full textual descriptions of the objects. (The latter should either be included in the Document collection or cited as published works in peer-reviewed scientific journals. See Chapter 7 for more details.)

File or Directory Name	Class	Cardinality	Name Fixed?
context[_✶]	–	0..✶	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1[2]	fixed
agency_1.0.tab	Product_Context	0..✶	prefix fixed
facility_1.0.tab	Product_Context	0..✶ [2]	prefix fixed
instrument_*_1.0.xml	Product_Context	0..✶ [3]	prefix fixed
instrument_host_*_1.0.xml	Product_Context	0..✶ [3]	prefix fixed
investigation_*_1.0.xml	Product_Context	0..✶ [3]	prefix fixed
node_*_.xml	Product_Context	1..✶	prefix fixed
other_*_1.0.xml	Product_Context	0..✶	prefix fixed
pds_affiliate_*_1.0.xml	Product_Context	0..✶	prefix fixed
pds_guest_*_1.0.xml	Product_Context	0..✶	prefix fixed
resource_*_1.0.xml	Product_Context	0..✶	prefix fixed
target_*_1.0.xml	Product_Context	0..✶ [3]	prefix fixed
telescope_*_1.0.xml	Product_Context	0..✶ [3]	prefix fixed

[2]Note that an inventory table listing primary context products should only exist in the PDS Engineering Node-curated PDS-wide context collections. Thus, the collection_primary_1.0.tab file should never appear in archive bundles delivered by data providers to the PDS.

[3]Note that these products are required in cases where the corresponding *physical* or *conceptual* object exists and is associated with data in the archive bundle.

Data preparers should send primary context products directly to a PDS Engineering Node data engineer or their respective Discipline Nodes. PDS personnel will include these products as primary members of a PDS-wide context collection.

Since they are primary members of a PDS-wide collection, they are only to be included in archive bundles as secondary members of a context collection. For this reason, a context collection is not required for inclusion in bundles delivered by data providers to the PDS. (However, PDS nodes are required to include copies of these files in a context collection as part of a bundle delivered to the deep archive.)

2.6 Data Directory

Each data directory contains a data collection. Data collections contain basic observational products.

For regular science archives, the data directory is required. For special archives (like a SPICE bundle or a system bundle), the directory is optional.

File or Directory Name	Class	Cardinality	Name Fixed?
data[_✶]	–	0..✶	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
orbits_0001_0045	–	1..✶	not fixed
c194306nnn	–	1..✶	not fixed
c194306019.xml	Product_Observational	1..✶	not fixed
c194306019.tab	–	1..✶	not fixed
⋮	⋮	⋮	⋮
c194307nnn	–	1..✶	not fixed
c194307053.xml	Product_Observational	1..✶	not fixed
c194307053.tab	–	1..✶	not fixed
⋮	⋮	⋮	⋮

Each data directory may contain multiple levels of sub-directories. The structure of these subdirectories is at the discretion of the data provider.

Observational products shall be implemented using the Product_Observational class.

2.7 Document Directory

Each document directory contains a single document collection. A document collection shall include all documents deemed useful by the data preparer and cognizant discipline node to the science investigator in facilitating the understanding, interpretation, and use of the data included in the archive bundle. A detailed specification for the archive's observational data products is usually considered to be an essential document. Examples of other types of documents that may be included in the document directory include:

archive volume specifications
calibration reports

data product specifications
derivation and analysis techniques
experiment results
functional requirements documents
instrument descriptions
mapping descriptions and results
mission descriptions
peer review reports
project documents
project web sites
software descriptions
spacecraft or telescope descriptions
user's manuals

Note, copies of published papers may be included in PDS archives only if the primary publication journal permits it under their copyright rules.[4]

File or Directory Name	Class	Cardinality	Name Fixed?
document[_✶]	–	0..✶	fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
archive_volume_sis	–	*sample*	not fixed
archive_volume_sis.pdf	–	1..✶	not fixed
archive_volume_sis.xml	Product_Document	1..✶	not fixed
calibration_report	–	*sample*	not fixed
calibration_report.doc	–	0..✶	not fixed
calibration_report.pdf	–	1..✶	not fixed
calibration_report.xml	Product_Document	1..✶	not fixed
data_product_sis	–	*sample*	not fixed
data_product_sis.html	–	0..✶	not fixed
data_product_sis.pdf	–	1..✶	not fixed
data_product_sis.xml	Product_Document	1..✶	not fixed
data_product_sis_app.html	–	1..✶	not fixed
data_product_sis_intro.html	–	1..✶	not fixed
data_product_sis_proc.xml	–	1..✶	not fixed
data_product_sis_struct.xml	–	1..✶	not fixed
images	–	0..✶	not fixed
image_01.jpg	–	0..✶	not fixed
image_02.jpg	–	0..✶	not fixed
image_03.jpg	–	0..✶	not fixed
⋮	⋮	⋮	⋮

[4]Note that *Space Sciences Review* permits inclusion of copies of these documents in PDS archives.

2.8 Geometry Directory

Each geometry directory contains a single geometry collection. Geometry collections contain non-SPICE geometry products (for example, Supplementary Experiment Data Record (SEDR) data or gazetteers). Detailed information about particular cartographic projections utilized in the archive may be included here.

File or Directory Name	Class	Cardinality	Name Fixed?
geometry	–	0..1	fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
gazetteer.tab	–	1..✶	not fixed
gazetteer.xml	TBD	1..✶	not fixed
⋮	⋮	⋮	⋮

2.9 Miscellaneous Directory

The miscellaneous directory contains supplemental meta-data catalogs, database dumps, indices, errata files, or spreadsheets deemed by the data provider to be useful to the interpretation of the data in the archive. Any updates made to product label values after the archive is ingested may be included in the miscellaneous directory in tabular form. Examples of the types of information contained in the miscellaneous directory include:

database dumps
errata
footprint files
modification history
tables of anaglyph pairs
updates

File or Directory Name	Class	Cardinality	Name Fixed?
miscellaneous[_✶]	–	0..✶	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
⋮	(as appropriate)	⋮	⋮

The miscellaneous collection shall be implemented using a Product_Collection class.

Because of the disparate nature of the files included in the miscellaneous directory, no single Product class exists for miscellaneous products. Any of the following are available for use: Product_Document, Product_File_Text, Product_Observational, Product_Thumbnail, Product_Update, or Product_Zipped.

2.10 SPICE Kernels Directory

Each spice_kernels directory contains a SPICE collection. The SPICE collections in a PDS bundle contain individual SPICE kernel files and their XML labels.

File or Directory Name	Class	Cardinality	Name Fixed?
spice_kernels_*	–	0..∗	prefix fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
ck	–	0..1	fixed
dbk	–	0..1	fixed
dsk	–	0..1	fixed
ek	–	0..1	fixed
fk	–	0..1	fixed
ik	–	0..1	fixed
lsk	–	0..1	fixed
mk	–	0..1	fixed
pck	–	0..1	fixed
sclk	–	0..1	fixed
spk	–	0..1	fixed

SPICE collections may contain SPICE kernel files produced by the NAIF node in conjunction with a flight project, SPICE files produced by individual data providers, or a mixture of both. In cases where non-NAIF data providers are including NAIF-produced SPICE products in their archives, these products should be included as secondary members of their respective SPICE collections.

Kernel files and their XML labels must be placed in the following subdirectories based on the kernel type:

ck	CK files (spacecraft and instrument orientation data)
dbk	DBK files (databases in SPICE format)

dsk	DSK files (digital shape data for natural bodies)
ek	EK files (events information)
fk	FK files (reference frames definitions)
ik	IK files (instrument parameters and FOV definitions)
lsk	LSK files (leapsecond information)
mk	MK files (meta-kernels listing kernels to be used together)
pck	PCK files (natural body rotation and size/shape constants)
sclk	SCLK files (spacecraft clock correlation data)
spk	SPK files (trajectory and ephemeris data)

Note that SPICE kernel files have specific naming requirements. In particular, there are mandated file extensions based on the type of kernel file. A list of these reserved file extensions is available in Table 6.2.

2.11 XML Schema Directory

Note that there is only one XML schema collection permitted per archive bundle. The xml_schema directory shall contain this single schema collection. The schema collection shall contain all XML schema files included in or referenced by XML labels in the archive along with any Schematron files created for validation purposes. An XML catalog file shall also be included.

File or Directory Name	Class	Cardinality	Name Fixed?
xml_schema	–	1	fixed
collection_1.0.xml	Product_Collection	1	fixed
collection_1.0.tab	–	1	fixed
products	–	1	fixed
catalog.xml	–	1	fixed
catalog_label.xml	Product_XML_Schema	1	fixed
PDS4_PDS_0300a.xml	Product_XML_Schema	1	fixed[5]
PDS4_PDS_0300a.xsd	–	1	fixed[5]
PDS4_PDS_0300a.sch	–	1	fixed[5]
mission_dictionary_1.0.xml	Product_XML_Schema	1	not fixed
mission_dictionary_1.0.xsd	–	1	not fixed
node_dictionary_1.0.xml	Product_XML_Schema	1	not fixed
node_dictionary_1.0.xsd	–	1	not fixed
specific_product_1.0.sch	–	1	not fixed
specific_product_1.0.xml	Product_XML_Schema	1	not fixed
⋮	⋮	⋮	⋮

[5]The name of this file will vary depending on the version of the information specification used by the data preparer; however, the file, as received from the PDS, should not be renamed.

A schema collection will typically include both *primary* and *secondary* members. Since the PDS-wide schema files are registered as part of a PDS system collection, these files are included here as *secondary* members of this collection.

Chapter 3

Labels

PDS product labels are required for describing the contents and format of each individual product within an archive. Specifically, the labels contain the *description objects* that describe corresponding *data objects*. (See Appendix A.1, "Introduction", in the *Concepts Document* (PDS, 2012a) for a more detailed explanation.)

The description objects are populated using a standard set of classes, attributes, and standard values. These classes, attributes, and standard values constitute the *PDS4 Information Model* (PDS, 2012c) and are defined in the *PDS4 Data Dictionary* (PDS, 2012b).

The PDS4 Information Model is expressed, and therefore PDS labels are written, in the eXtensible Markup Language (XML).

In order for any document (including PDS labels) to meet the XML standard, it must be both "well formed" and "valid". A well formed document must have correct XML syntax; a valid document must conform to the rules of an XML schema document (XSD). XSDs provide the rules governing the structure and content of a specific XML document class. The PDS currently splits these latter two functions between XSDs (structure) and Schematron schemas (content).

Thus, in order for a label to be validated as complying with PDS4 standards, it must:

- have correct XML syntax
- be compliant with the class and attribute structures defined by the PDS and relevant discipline nodes and missions in their respective XSDs
- be compliant with the rules governing specific attributes and their values as set by the PDS and relevant discipline nodes and missions in their respective Schematron schemas

PDS4 schemas are supplied to data providers by the PDS. Missions and other data providers may not modify these existing schemas; however, they may extend existing classes and provide their own additional attributes in their own dictionary schemas, with the approval of a PDS discipline node.

Under the PDS4 standard, all product labels are detached from the digital files containing the data objects they describe. There is one PDS4 label for every product. Each product may contain one or more data objects. The data objects of a given product may all reside in a single file, or they may be stored in multiple separate files. PDS4 label files must end with the file extension ".xml".

Figure 3.1 shows the general structure of the label of an observational product. Most PDS products utilize some or all of the same components in their labels. (A complete listing of which label components are present in which products is shown in Table 3.1.) The following sections, organized in accordance with this overarching structure, provide general descriptions of the classes used in PDS4 labels, plus details about variations seen in specific types of products.

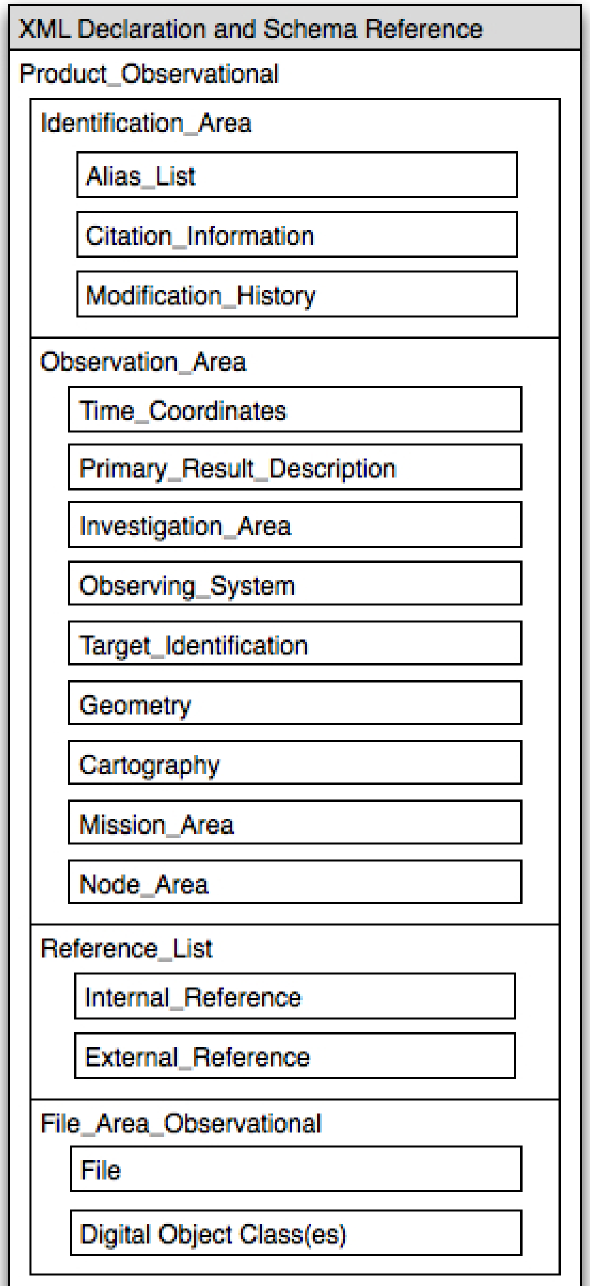


Figure 3.1: Label structure of a generic observational product

3.1 The XML Declaration and Schema Reference

While the preamble to a PDS label will typically be its shortest section, it is important that it be populated correctly or validation of the label may be rendered impossible.

Figure 2.2 in the previous chapter shows an example of a simple label preamble. The first line of any XML document is an XML declaration:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Label Component	**Product_Browse**	**Product_Bundle**	**Product_Collection**	**Product_Context**	**Product_Document**	**Product_File_Text**	**Product_Observational**	**Product_SPICE_Kernel**	**Product_Thumbnail**	**Product_Update**	**Product_XML_Schema**
XML Declaration and Schema Reference	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Identification Area	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Context Area / Observation Area	✗	C	C	✗	C	✗	O	O	✗	✗	✗
Discipline Area	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Reference List	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Bundle	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
Collection	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
Context Object	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Document Description	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Update	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
File Area	✓	✓	✓	✗	✗	✓	✓	✓	✓	✗	✓
Document Format Set	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗
Bundle Member Entry	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗

Table 3.1: Label components based on product type Needs updating.

This line declares the document to be an XML version 1.0 document using the `UTF-8` encoding. (Note that although other encodings are permissible in XML documents, PDS only permits `ASCII` or `UTF-8`.)

Immediately following the XML declaration statement are the processing instructions. These lines report to a parsing application how to find the validation files for the label. The instructions take the form of:

```
<?xml-model href="schema loc" schematypens="schema namespace"?>
```

where *`schema loc`* is a Uniform Resource Identifier (URI) providing the location of the validation schema and *`schema namespace`* is a namespace indicating the type of schema language the referenced schema is written in.

Every PDS4 label must be validated against the PDS master Schematron file. The URI for the current version of this file is:

```
http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch
```

The namespace for the Schematron validation language is:

```
http://purl.oclc.org/dsdl/schematron
```

Thus, every PDS4 label should contain the following lines:

```
<?xml-model
  href="http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch"
  schematypens="http://purl.oclc.org/dsdl/schematron"?>
```

Any additional, node- or mission-specific Schematron validation files should be referenced the same way. However, it is important to remember when you are building product labels that you should not be including the *current* location of validation files, but what you expect to be their permanent location once your archive is complete and registered with the PDS. (See section 6.1, *Namespaces*, for information on how to determine the permanent location of your schema files.) In the meantime, a catalog file can be used to map the permanent location listed in the label to a working location on your file system. (See Appendix **??**, **??**.)

After the processing instructions follows a line declaring the root element of the XML document. In a PDS4 label, this should identify the type of product the label is describing and should be one of the following:

- `Product_Browse` - a basic product containing a low resolution or "quick-look" version of an observational product

- `Product_Bundle` - an aggregate product used to identify the member collections of an archive bundle
- `Product_Collection` - an aggregate product used to identify the member basic products of an archive collection
- `Product_Context` - a basic product identifying the *physical* (instrument, spacecraft, target, people) and *conceptual* (investigation, node) objects related to an observational product's provenance
- `Product_Document` - a basic product identifying a single logical document, such as an interface specification, instrument description, or user's manual; the document product may comprise multiple formats
- `Product_File_Text` - a basic product consisting of a single digital file with ASCII character encoding
- `Product_Observational` - a basic product comprising images, tables, and other fundamental data structures that are the result of a science or engineering investigation
- `Product_SPICE_Kernel` - a basic product consisting of a SPICE kernel
- `Product_Thumbnail` - a basic product consisting of a highly reduced version of an observational product, typically used in displaying the results from search interfaces
- `Product_Update` - a basic product containing information about updates to observational products that have already been archived
- `Product_XML_Schema` - a product consisting of XML formatted schemas, Schematron files, OASIS catalog files, or any other reference schemas used in the interpretation of an observational product

Immediately following the root element, as part of the same XML statement, should be a list of the namespaces which are used in the label. A simple label, like that for the Product_Bundle, will only need the `pds` namespace (in which the PDS classes and attributes are defined) and the `xsi` namespace, in which the XML elements themselves are defined. More complex labels will need to refer to additional namespaces for PDS discipline nodes and missions.

Namespaces in an XML document are indicated by a namespace abbreviation, followed by a colon, followed by an XML element or attribute. Note, for example, the `xsi:schemaLocation` shown on line 10 of Figure 2.2 and line 15 of Figure 3.2. The `xsi` is an abbreviation for the *XML Schema Instance* namespace formally designated by the URI:

```
http://www.w3.org/2001/XMLSchema-instance
```

The `schemaLocation` is an XML attribute of the root element and is defined in the `xsi` namespace.

XML documents may have a default namespace. In this case, bare elements and attributes (i.e., those that are not pre-fixed with a namespace abbreviation followed by a colon) are assumed to belong to the default namespace. All PDS4 labels should have the `pds` namespace set as the default namespace.

The listing of the namespaces which follows the root element provides a mapping between the namespace abbreviations and the namespace URIs. The first of these lines does not include a namespace abbreviation and indicates the default namespace:

```
xmlns="http://pds.nasa.gov/pds4/pds/v03"
```

The remaining lines specify namespace abbreviations. It is generally considered a good practice to repeat the `pds` namespace, this time with an explicit abbreviation, in case it is needed somewhere in the label or dictionaries to resolve an ambiguity. The `xsi` namespace must also be listed, as well as any others needed in the label. Consult your PDS discipline node (or see section 6.1) to learn which namespaces to use in your labels.

```
xmlns:pds="http://pds.nasa.gov/pds4/pds/v03"
xmlns:img="http://pds.nasa.gov/pds4/img/v1"
xmlns:mpf="http://pds.nasa.gov/pds4/mission/mpf/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

Finally, the `xsi:schemaLocation` attribute of the root element provides hints to a parsing application as to how to locate the schemas used to assess the validity of the label. The `xsi:schemaLocation` contains pairs of values which map each namespace to the URI for a schema which defines that namespace. Thus, in the example below, the `pds` namespace is shown as the first value, and the URI for the PDS4 master schema, which defines all the PDS classes and attributes for that namespace, is listed second. Remember that a URI is an identifier, not a location, although it *may* (and in this case *does*) resolve to a real location. A catalog file (see Appendix **??**) can be used to map this URI to an actual, physical location, either on the Internet or on your local machine.

```
xsi:schemaLocation="
   http://pds.nasa.gov/pds4/pds/v03
   http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.xsd">
```

The Product_Bundle example shown in Figure 2.2 has no dependencies outside of the master PDS schema and Schematron files, and so only references those two files. A more complex example is shown in Figure 3.2. This is the header of an observational product that references both a node dictionary (`imaging_dictionary.xsd`) and a mission-specific dictionary (`mpf_dictionary.xsd`). Note that both the namespaces and the URIs of these files are provided. The label also references a product-specific Schematron validation file (`mpf_imp_raw.sch`).

3.2 The Identification Area

The purpose of the Identification_Area in a PDS label is to uniquely identify the product the label describes. Since this identifier is used to track the product in the PDS registry, every PDS product

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
    href="http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>
<?xml-model
    href="http://pds.nasa.gov/pds4/mission/mpf/v1/mpf_imp_raw.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>

<Product_Observational
    xmlns="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:pds="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:img="http://pds.nasa.gov/pds4/img/v1"
    xmlns:mpf="http://pds.nasa.gov/pds4/mission/mpf/v1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://pds.nasa.gov/pds4/pds/v03
        http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.xsd

        http://pds.nasa.gov/pds4/img/v1
        http://pds.nasa.gov/pds4/img/v1/imaging_dictionary.xsd

        http://pds.nasa.gov/pds4/mission/mpf/v1
        http://pds.nasa.gov/pds4/mission/mpf/v1/mpf_dictionary.xsd">
```

Figure 3.2: The XML declaration and schema reference for an observational product label with node and mission dictionaries and a mission validation file.

must have an Identification_Area.

An example of an Identification_Area is shown in Figure 3.3.

The first two attributes of the Identification Area are the `logical_identifier` and the `version_id`. Together these constitute the product's LIDVID, an identifier for this product that is globally unique. The LIDVID is used to ingest the product into the PDS registry and is used by other products to provide a reference to this product. For example, a browse version of this image would indicate its full resolution source image by providing the LIDVID of the source product. (For details on how to construct LIDVIDs and other identifiers, consult sections 6.2 and 6.3 of this document.)

The `title` attribute is a name for the product. It is a simple text string, with no specific format requirements. It is not necessary for the title to be unique for each product. The title will typically be displayed along with the product in a product display tool.

```xml
<Identification_Area>
    <logical_identifier>urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1247554985-REGULAR-0164020283</logical_identifier>
    <version_id>2.0</version_id>
    <title>GALLERY_PAN/PRESIDENTIAL_PAN_TIER_4_GREEN.1CMD</title>
    <information_model_version>0.3.0.0.a</information_model_version>
    <product_class>Product_Observational</product_class>
    <Alias_List>
        <Alias>
            <alternate_id>IMP_EDR-1247554985-REGULAR-016020283</alternate_id>
            <comment>PDS product_id</comment>
        </Alias>
        <Alias>
            <alternate_id>16020283</alternate_id>
            <comment>PDS image_id</comment>
        </Alias>
    </Alias_List>
    <Modification_History>
        <Modification_Detail>
            <modification_date>2012-01-18</modification_date>
            <version_id>2.0</version_id>
            <description>This product has been migrated to PDS4 from an original PDS3
                product.</description>
        </Modification_Detail>
    </Modification_History>
</Identification_Area>
```

Figure 3.3: A sample Identification_Area for an observational product.

The information_model_version indicates the version of the PDS Information Model with which this product complies. The value of this attribute may be discovered in a couple of different places. The third line of the PDS master schema is a comment, indicating the version of the PDS Information Model. The version shown there, excluding the initial "V", provides the correct value for this attribute. Alternatively, a text search of the master Schematron schema on the string "information_model_version" will show the correct value expected for the attribute. For the current version of the model, this attribute should be populated with the value "0.3.0.0.a".

The product_class attribute provides the name of the product class used to describe the product. This information is used by the PDS registry to assist in product tracking and reporting. The value of this attribute should be the same as the name of the root element of the product and should correspond to one of the values listed in section 3.1 above.

There is one exception to this rule. Context products should not have a product_class of "Product_Context", but rather should contain a pseudo-product class constructed from the word Product_ and the name of the context object that the label describes. Thus, the product_class for a context product should have one of the following values:

- Product_Instrument
- Product_Instrument_Host
- Product_Investigation
- Product_Node
- Product_Other
- Product_PDS_Affiliate
- Product_PDS_Guest
- Product_Resource
- Product_Target

These five required attributes constitute the core of the Identification Area. In addition to these, there are three optional classes which can be used to provide additional identifying information.

3.2.1 Alias List

The Alias_List class is an optional class that provides a method for specifying additional identifiers by which the product has been known in the past, or which may be used to locate the product in other data systems.

The Alias_List class contains one or more Alias classes. Each Alias class provides a single alternative method for identifying a product. It must contain at least one of an alternate_id or an alternate_title. (It may also contain both, as long as they are associated with one another. However, unrelated ids and titles should not be mixed within a single Alias class.) The `comment` attribute is used to provide a brief explanation of the nature or context of the alternative identifier.

```
<Alias_List>
   <Alias>
      <alternate_id>IMP_EDR-1247554985-REGULAR-0164020283</alternate_id>
      <comment>PDS3 product_id</comment>
   </Alias>
   <Alias>
      <alternate_id>164020283</alternate_id>
      <comment>PDS3 image_id</comment>
   </Alias>
</Alias_List>
```

The PDS registry-based search engine will support searches for products using these alternative identifiers.

3.2.2 Citation Information

The Citation_Information class is used to provide the necessary information to build a proper citation to the product that the label describes. It is also used to populate the Astrophysics Data System (ADS) abstract service. This class is optional.

Citations to PDS products should take the form of:

```
<author1, author2, and author3>, <product logical identifier>,
<publisher>, <year of publication>
```

The logical identifier of a product is supplied by the `logical_identifier` attribute of the Identification_Area class. The publisher of a PDS product is always "NASA Planetary Data System". The Citation_Information class supplies supplementary citation information not found elsewhere in the label.

If you wish to create a citation to a PDS basic product that does not contain a Citation_Information class, use the `logical_identifier` for the basic product and the citation information provided in the closest parent aggregate product (i.e., collection or bundle) that does contain the information.

When the product for which the citation information is being provided is a document, the population of the fields is relatively self-explanatory:

```
 <Citation_Information>
    <author_list>Britt, D.; Maki, J.; Weinberg, J.</author_list>
    <publication_year>1997</publication_year>
    <keywords>Imager for Mars Pathfinder, users guide</keywords>
    <description>
This User's Guide for the Imager for Mars Pathfinder (IMP) provides an
overview of the camera system and optics, calibration, data compression,
instrument operating parameters, and commands.</description>
 </Citation_Information>
```

Note that the `description` attribute has a data type of `ASCII_Text_Preserved`, which means that white space in the value is preserved. Thus, even though line breaks were included in the above example for the sake of readability, you should not include any line breaks in the text for this attribute in your product labels, unless you explicitly want those line breaks preserved (for example, between paragraphs).

In a Citation_Information class for an observational product or data collection, the individuals included in the `author_list` can be somewhat more ambiguous. Generally the principal investigator for the instrument responsible for collecting the data, as well as the person primarily responsible for producing the data, should be included. However, this is fundamentally up to the discretion of the data production team.

The `editor_list` should include the name(s) of the PDS personnel responsible for interfacing with the data provider, selecting peer reviewers, assessing liens against the data, and formally approving publication of the data with the PDS.

```
 <Citation_Information>
    <author_list>Smith, P.; Runkle, A. J.</author_list>
    <editor_list>Duxbury, E. D.</editor_list>
    <publication_year>1998</publication_year>
    <keywords>Mars Pathfinder, Mars, Imager for Mars Pathfinder,
       Mars Pathfinder Lander</keywords>
    <description>
This collection contains images taken by the Imager for Mars Pathfinder
on July 4 through September 27, 1997 (plus a few pre-landing calibration
images).  The images have been decoded and decompressed in single
frame form, but not calibrated or radiometrically corrected.</description>
 </Citation_Information>
```

A sample citation built using the the information from the Citation_Information class shown above would be:

```
Smith, P. and A. Runkle,
urn:nasa:pds:mars_mpf_imp_raw:data:IMP_EDR-1247554985-REGULAR-0164020283,
NASA Planetary Data System, 1998.
```

3.2.3 Modification History

The Modification_History class provides details about changes to a product once it enters the PDS registry. There should be one Modification_Detail class for every version of a product after the first registered product.

```
  <Modification_History>
     <Modification_Detail>
        <modification_date>2012-01-18</modification_date>
        <version_id>2.0</version_id>
        <description>
This product has been migrated to PDS4 from an original PDS3 product.</description>
     </Modification_Detail>
  </Modification_History>
```

The `modification_date` attribute specifies the date that the relevant version of the product was modified. (This may not correspond to an actual file system modification date, since updated versions of products are frequently re-generated, rather than modified. In this case, the file system creation date of the new file should be used to populate the `modification_date` attribute.)

The `version_id` attribute corresponds to the `version_id` attribute of the Identification_Area class.

Note that the Modification_History is not meant to provide a complete processing history of the product. That information, if desired, is currently stored within the node or mission areas of the label.

3.3 The Observation Area / Subject Area

The next area of the label provides descriptive parameters for the product. This does not include the physical This includes such diverse information as the time and location of the event, the instrument

The "Observation_Area" of the label is the area where the observation parameters are listed. The top portion of this area includes a relatively small set of attributes that are common across much of the PDS archive. These are attributes that are deemed to be widely enough used that it makes sense for them to have common definitions across all PDS disciplines. More specific classes and attributes are located in the "Mission_Area" and "Node_Area" (see below).

```
<Observation_Area>
  <comment>comment</comment>
  <Time_Coordinates>
  ...
  </Time_Coordinates>

  <Primary_Result_Description>
  ...
  </Primary_Result_Description>

  <Investigation_Area>
  ...
  </Investigation_Area>

  <Observing_System>
  ...
  </Observing_System>

  <Target_Identification>
  ...
  </Target_Identification>

  <Geometry>
  ...
  </Geometry>
  <Cartography>
  ...
  </Cartography>
  <Mission_Area>
  ...
  </Mission_Area>
  <Node_Area>
  ...
  </Node_Area>
</Observation_Area>
```

3.3.1 Time Coordinates

```
<Time_Coordinates>
   <start_date_time>1997-07-07T05:13:42.763Z</start_date_time>
   <stop_date_time xsi:nil="true"/>
   <local_true_solar_time>13:39:12</local_true_solar_time>
</Time_Coordinates>
```

3.3.2 Primary Result Description

```
<Primary_Result_Description>
  <type>Image</type>
  <purpose>Science</purpose>
  <data_regime>Visible</data_regime>
  <reduction_level>Raw</reduction_level>
</Primary_Result_Description>
```

3.3.3 Investigation Area

```
<Investigation_Area>
  <name>MARS PATHFINDER</name>
  <type>Mission</type>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:investigation.MARS_PATHFINDER</lid_reference>
    <reference_type>data_to_investigation</reference_type>
  </Internal_Reference>
</Investigation_Area>
```

3.3.4 Observing System

```
<Observing_System>
  <Observing_System_Component>
    <name>MARS PATHFINDER LANDER</name>
    <observing_system_component_type>Spacecraft</observing_system_component_type>
    <Internal_Reference>
      <lid_reference>urn:nasa:pds:instrument_host.MPFL</lid_reference>
      <reference_type>is_instrument_host</reference_type>
    </Internal_Reference>
  </Observing_System_Component>
</Observing_System>
```

3.3.5 Target Identification

```
<Target_Identification>
</Target_Identification>
```

3.3.6 Geometry

TBD

3.3.7 Cartography

TBD

3.3.8 The Mission Area

The “Mission_Area” of the label provides the container for all mission-specific classes and attributes. These classes and attributes are defined in a local data dictionary (see section **??**), and must be prefixed with the namespace identifier applicable to that local dictionary.

```
<Mission_Area>
  <mpf:application_packet_id>34</mpf:application_packet_id>
  <mpf:application_packet_name>SCI_IMG_3</mpf:application_packet_name>
  <mpf:auto_exposure_data_cut>3000</mpf:auto_exposure_data_cut>
  <mpf:auto_exposure_pixel_fraction>1.0000</mpf:auto_exposure_pixel_fraction>
  <mpf:frame_id>BOTH</mpf:frame_id>
</Mission_Area>
```

3.3.9 The Node Area

The “Node_Area” of the label provides the container for all discipline-specific classes and attributes. These are the classes and attributes that are defined in various PDS node-level local data dictionaries (see section **??**); they must be prefixed with the relevant node’s namespace identifier.

```
<Node_Area>
  <img:Camera_Parameters>
    <img:exposure_duration>46.0</img:exposure_duration>
    <img:exposure_type>AUTO</img:exposure_type>
    <img:filter_name>L670_R670</img:filter_name>
    <img:filter_number>5</img:filter_number>
  </img:Camera_Parameters>
</Node_Area>
```

Note that it is possible to utilize classes and attributes pulled from multiple node dictionaries. In this case, a separate Node_Area would be used for each node's elements.

Schemas and dictionaries for node level classes and attributes are currently only available in the Information Model Specification and in the http://pds.nasa.gov/schema/pds4/generic/[node] directories. They may be included as an appendix in a future version of this document.

3.4 The Reference List

3.5 The File Area

The "File_Area" of a label is used to describe the system files containing the digital objects described in the "Data_Area". It must contain a distinct File class for each file containing data for this product.

Each File class must be described by a local_identifier, unique within the product label. This identifier will be used in the Data_Location class (described below) to tie together a Digital Object class with the digital data it describes. Local_identifiers may be re-used from one label to another within a collection.

```
<File_Area_Observational>
  <File>
    <local_identifier>FILE</local_identifier>
    <creation_date_time>1998-07-14T00:36:08.000Z</creation_date_time>
    <file_name>i943630r.img</file_name>
    <file_size>138240</file_size>
    <max_record_bytes>512</max_record_bytes>
    <md5_checksum>fa1db0cfb1cea71e438ab791a6ee766d</md5_checksum>
    <records>270</records>
  </File>
  <Array_2D_Image base_class="Array_Base">
    ...
  </Array_2D_Image>
  <Header>
    ...
  </Header>
</File_Area_Observational>
```

3.5.1 The Digital Object Classes

3.6 The Closing Tag

In order to for the XML label to be well formed, it must end with a closing tag for the class opened at the beginning of the label (see section 3.1, above).

```
</Product_Observational>
```

Chapter 4

Fundamental Data Structures

There are four fundamental data structures that may be used for archiving data in the PDS. All data products delivered to the PDS must be constructed from one or more of these structures. These four fundamental structures are described using four base classes: Array (used for homogeneous n-dimensional arrays of scalars), Table_Base (used to describe repeating records of heterogeneous scalars), Parsable_Byte_Stream (a stream of bytes that can be parsed using standardized rules), and Encoded_Byte_Stream (an encoded stream of bytes). All other digital object classes in the PDS are derived from one of these four base classes.

4.1 Arrays

The first of the four basic PDS4 structures is the Array.[1] Any data structure which consists of fixed-length rows of homogeneous elements in any number of dimensions shall be described using the Array class or one of its subclasses. As two- and three-dimensional data structures are the most commonly used in science data, the following discussion will focus on them. However, the principles described below can be extrapolated to structures with greater numbers of dimensions.

4.1.1 Storage Order and Index Order

Using standard matrix notation, the row number of a particular element in an array is specified using the first index and the column number is specified using the second index. Thus, in the array

[1] Most of section 4.1 is adapted from the white paper by Showalter, 2012, with permission.

shown at left, the position of the “2” (row 1, column 2) is specified using the notation (1,2), while the position of the “4” (row 2, column 1) is specified as (2,1).

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

An N-dimensional array is always stored in computer memory or in a data file as a linear sequence of numbers. There are two ways to store array data in linear memory: row-major order and column-major order. In row-major storage, the rows of the array are stored sequentially. In column-major storage, the columns are stored sequentially.

Thus, using the above notation, a row-major 2x3 array would be stored this way in memory:

(1,1), (1,2), (1,3), (2,1), (2,2), (2,3)

or, using the numbers in the example above, it would be stored as follows:

1 2 3 4 5 6

The same array, in column-major order, would be stored this way:

(1,1), (2,1), (1,2), (2,2), (1,3), (2,3)

or:

1 4 2 5 3 6

PDS uses the terminology *Last_Index_Fastest* as a synonym for row-major order and *First_Index_Fastest* as a synonym for column-major order.

Last_Index_Fastest (row-major) is the required storage order for PDS4 array data. (Note that this fact does not require modifying data stored in First_Index_Fastest or column-major order; only the order of the axes in the label needs to be switched. See the discussion below under Axis Meaning.)

Note that FORTRAN, IDL, and MATLAB arrays are stored in First_Index_Fastest (column-major) order; all other major programming languages store arrays in Last_Index_Fastest (row-major) order.

Both storage orders were permitted in PDS3 data, but Last_Index_Fastest (row-major) was the default and recommended storage order.

FITS format uses First_Index_Fastest storage order if one assumes that the shape of the array is defined by (NAXIS1, NAXIS2, ...). However, the popular FITS programming libraries cfitsio and pylab describe the arrays as Last_Index_Fastest, i.e, with the shape (..., NAXIS2, NAXIS1).

VICAR indices are listed in First_Index_Fastest order, although this only affects the definitions of keywords N1, N2, N3, and N4. In practice, VICAR mixes together the issues of storage order and axis meaning, using other keywords NS, NL, NB, and ORG.

ISIS exclusively uses First_Index_Fastest storage order.

The Array class and its subclasses are used to describe array data.

Along with the fact that all array data is assumed to be binary, the above information is contained in the PDS label using the following attributes:

```
<Array_2D_Image>
    <local_identifier>IMAGE</local_identifier>
    <offset unit="byte">11264</offset>
    <axes>2</axes>
    <axis_index_order>Last_Index_Fastest</elements>
    <encoding_type>Binary</encoding_type>
    <Element_Array>
        ...
</Array_2D_Image>
```

offset Provides the starting location, within a file, of the stream of bytes constituting the data object; the first byte of the file has an offset of zero.

axes Indicates the number of dimensions of the array. This value is set to "2" for the Array_2D class and its subclasses, and to "3" for the Array_3D and its subclasses. Higher order arrays should use the generic "Array" class.

axis_index_order Specifies the storage order of the array in linear memory. In PDS4, it is set to "Last_Index_Fastest", which is synonymous with row-major storage order.

encoding_type Indicates binary versus character data and is always set to "Binary" for PDS4 arrays.

4.1.2 Axis Meaning

In a data product represented by an N-dimensional array, each axis needs to be assigned a meaning. Up to two axes can be spatial. Other common uses of axes are for RGB color, wavelength band, or a time-ordered sequence. The number of axes in a scientific data file can exceed three, but rarely does.

PDS uses the terms *Line* and *Sample* to distinguish the two spatial axes of an array. (It shares this usage with VICAR and ISIS, but not FITS.) Their relationship is defined exclusively by the storage order in the file, with the line number increasing more slowly than the sample number.

This means that using Last_Index_Fastest ordering, the first index would be the slowest varying or the Line dimension, while the second index would be the fastest varying or Sample dimension. For First_Index_Fastest ordering, the first index varies the fastest and would therefore contain the Sample dimension, while the second index varies slowest and would contain the Line dimension.

In other words, a 2-D image would have index order (Line, Sample) in Last_Index_Fastest ordering, or (Sample, Line) in First_Index_Fastest ordering.

In a data product with two spatial dimensions and one spectral dimension, PDS uses the term *Band* to designate the spectral axis.

Along with PDS3, FITS, VICAR, and ISIS, PDS4 places no restrictions on the order of axes with respect to their meanings, except as noted above, that the Sample axis is always designated as the faster varying of the two spatial axes, and the Line axis the slower. A spectral axis may vary faster than both spatial axes, slower than both, or in between the two. A data product in which the band axis varies slower than the two spatial axes is typically referred to as a *band sequential* file. A product in which the band axis varies faster than the two spatial axes is typically identified as *sample interleaved* or *band interleaved by pixel*. A product in which the band axis varies slower than the Sample axis, but faster than the Line axis, is referred to as *line interleaved* or *band interleaved by line*.

In PDS4, using Last_Index_Fastest storage order, a 3-D band sequential image or qube would have index order (Band, Line, Sample), a sample interleaved image would have index order (Line, Sample, Band), and a line interleaved image would have index order (Line, Band, Sample).

In order to maximize the efficiency of playing potentially large movie files, the temporal axis should be the slowest varying axis of any movie arrays.

The characteristics of each array axis are modeled using an Axis_Array class. There must be one Axis_Array class present in the label for each dimension of the array. The Axis_Array class has four attributes (of which three are shown in the example below):

```
    <Axis_Array>
        <axis_name>Sample</axis_name>
        <elements>32</elements>
       <sequence_number>2</sequence_number>
  </Axis_Array>
```

axis_name Identifies the meaning of each axis. This attribute is required. The value should be "Sample" for the fastest varying spatial axis and "Line" for the slowest varying spatial axis. For a single spectral axis, the value should be "Band". For a temporal axis (such as in a movie), the value should be "Time".

elements Indicates the number of elements in the array along each particular axis.

unit Indicates the units along a particular axis. (optional)

sequence_number Provides the mapping between the axes and the axis_index_order. The fastest varying axis should have the highest sequence_number; the slowest varying axis should have a sequence_number of 1.

4.1.3 Display Orientation

Properly defining the orientation in which an image file should be displayed on a display device is only important when a product has two spatial dimensions. Otherwise, the display order is somewhat arbitrary.

In a two-dimensional image, each spatial axis of the array is associated with a direction on a screen. In PDS4, as with most other major image standards, the Sample or fastest varying axis is associated with the horizontal direction on a screen. The Line, or slowest varying axis, is associated with the vertical direction.

All major science image formats use the convention that sample data should be displayed on a display device from left to right. PDS shares this convention. There is not, however, conformity in how the line dimension is handled. (FITS, by convention, displays lines from bottom to top of a screen; VICAR and ISIS display from top to bottom.) Therefore, PDS leaves the choice of line display direction up to the data provider.

This information is encoded in a PDS label using the Display_2D_Image class:

```
<Display_2D_Image>
    <line_display_direction>Down</line_display_direction>
    <sample_display_direction>Right</sample_display_direction>
</Display_2D_Image>
```

line_display_direction Indicates the direction that Line (slowest varying spatial) axis should be displayed. May be either “Down” (meaning increasing from top to bottom of a display device) or “Up” (meaning increasing from bottom to top of a display device).

sample_display_direction Indicates the direction that Sample (fastest varying spatial) axis should be displayed. Must always be “Right”, meaning increasing from left to right across a display device.

4.1.4 Element Characteristics

The characteristics of the homogeneous elements (or pixels) in an array are described using the Element_Array class. The Element_Array class has four attributes:

```
<Element_Array>
    <data_type>UnsignedMSB4</data_type>
    <unit>uW*cm**-2*sr**-1*um**-1</unit>
    <scaling_factor>1.0</scaling_factor>
    <value_offset>0.0</value_offset>
</Element_Array>
```

Figure 4.1: Sample Element_Array describing the pixels in the core of a Galileo NIMS qube.

data_type Describes the physical storage space required for each element in an array, as well as indicating the storage order and interpretation of the bytes. (For a detailed description of the permissible values of the data_type attribute and their meanings, see section 5.3 of this document.)

unit Provides the physical units in which the measured quantity is stored. (Details on units are available in Chapter 10.) (optional)

scaling_factor Provides a multiplier to be applied to a stored value in a table to recover the original observed value. Original value V_o is calculated from the stored value V_s thus: $V_o = (V_s * scaling\_factor) + value\_offset$. (optional)

value_offset A number to be added to a value stored in a table to recover the original observed value. (See scaling_factor.) (optional)

4.2 Table Base

The second of the four basic PDS4 structures is the Table_Base. Data structures which consist of fixed-length rows of heterogeneous elements shall be described using the Table_Base class or one of its subclasses. At the current time, only two-dimensional tables are permitted.

(Note that variable-length tables, or "spreadsheets" are described in PDS4 as a subclass of the Parsable_Byte_Stream and are documented starting in section 4.3.1.)

Conceptually, tabular data files consist of a series of named columns containing both data locations and data values. The data may consist of both numbers and text strings.

Physically, the data are stored as a sequence of repeating records where each record is terminated by record delimiter characters. Fields within each record are of fixed length and begin at fixed locations. Since both field lengths and record lengths are fixed, field values are identifiable by position alone. However, field delimiters may optionally be included.

The data may be represented in any of binary, ASCII, or UTF-8 values.

Binary tables must be described using the Table_Binary subclass of Table_Base, and ASCII and UTF-8 tables must be described using the Table_Character subclass. These two structures are similar, except that the former contains records described using a Record_Binary class and the latter contains records described using a Record_Character class. Because the overall structure and associations are the same, only the Table_Character will be discussed below.

```
<Table_Character>
    <offset unit="byte">0</offset>
    <records>11</records>
    <encoding_type>Character</encoding_type>
    <description>This table contains the composition of
        some of the APXS targets, expressed as
        oxides in weight percent, normalized to
        a sum of 98%.</description>
    <record_delimiter>carriage_return line_feed</record_delimiter>
    <Record_Character>
        ...
```

offset Provides the starting location, within a file, of the stream of bytes to be parsed; the first byte of the file has an offset of zero.
records Indicates the total number of records in the data object.
encoding_type Indicates the encoding of the data, whether "Binary" (for Table_Binary) or "Character" (for Table_Character).
description Provides a short textual description of the table data. (optional)

The Record_Character class is used to describe the structure of each record in the table.

```
    <Record_Character>
        <record_length unit="byte">115</record_length>
        <fields>24</fields>
        <Field_Character>
            ...
```

record_delimiter Provides a text description of the character or character sequence used as a delimiter between records in a table object. Valid values are `carriage_return`, `carriage_return line_feed`, and `line_feed`.
record_length Provides the length, in bytes, of each record in the data object, including the record delimiter character(s).
fields Indicates the number of fields in each record.

```
A-2  S After deployment  2.3 0.9 7.9 1.2  7.4 0.7 51.0 2.5 4.0  0.8 0.5 0.1 0.2 0.1 6.9 1.0 1.2 0.2 16.6 1.7
A-3  R Barnacle Bill     3.2 1.3 3.0 0.5 10.8 1.1 58.6 2.9 2.2  0.4 0.5 0.1 0.7 0.1 5.3 0.8 0.8 0.2 12.9 1.3
A-4  S Next to Yogi      3.8 1.5 8.3 1.2  9.1 0.9 48.0 2.4 6.5  1.3 0.6 0.2 0.2 0.1 5.6 0.8 1.4 0.2 14.4 1.4
A-5  S Dark next to Yogi 2.8 1.1 7.5 1.1  8.7 0.9 47.9 2.4 5.6  1.1 0.6 0.2 0.3 0.1 6.5 1.0 0.9 0.1 17.3 1.7
A-7  R Yogi              1.7 0.7 5.9 0.9  9.1 0.9 55.5 2.8 3.9  0.8 0.6 0.2 0.5 0.1 6.6 1.0 0.9 0.1 13.1 1.3
A-8  S Scooby Doo        2.0 0.8 7.1 1.1  9.1 0.9 51.6 2.6 5.3  1.1 0.7 0.2 0.5 0.1 7.3 1.1 1.1 0.2 13.4 1.3
A-10 S Next to Lamb      1.5 0.6 7.9 1.2  8.3 0.8 48.2 2.4 6.2  1.2 0.7 0.2 0.2 0.1 6.4 1.0 1.1 0.2 17.4 1.7
A-15 S Mermaid Dune      1.3 0.7 7.3 1.1  8.4 0.8 50.2 2.5 5.2  1.0 0.6 0.2 0.5 0.1 6.0 0.9 1.3 0.2 17.1 1.7
A-16 R Wedge             3.1 1.2 4.9 0.7 10.0 1.0 52.2 2.6 2.8  0.6 0.5 0.2 0.7 0.1 7.4 1.1 1.0 0.1 15.4 1.5
A-17 R Shark             2.0 0.8 3.0 0.5  9.9 1.0 61.2 3.1 0.7  0.3 0.3 0.2 0.5 0.1 7.8 1.2 0.7 0.1 11.9 1.2
A-18 R Half Dome         2.4 1.0 4.9 0.7 10.6 1.1 55.3 2.8 2.6  0.5 0.6 0.2 0.8 0.1 6.0 0.9 0.9 0.1 13.9 1.4
```

Figure 4.2: Portion of a fixed width table. (The table contains composition data in oxide weight percent for various targets at the Mars Pathfinder landing site. The data was acquired from the APXS instrument.)

The Field_Character class is used to describe the structure of each field in the records. There must be one Field_Character class for every field in the record.

```
<Field_Character>
    <name>Sample Name</name>
    <field_number>1</field_number>
    <field_location unit="byte">1</field_location>
    <data_type>ASCII_Short_String_Collapsed</data_type>
    <field_length unit="byte">4</field_length>
    <field_format>%-4s</field_format>
    <description>This is the designation assigned to a
        particular APXS investigation site.</description>
</Field_Character>
```

name Provides a short text string by which a field is known.
field_number Provides the position of a field within a series of fields, counting from 1. (optional)
field_location Indicates the starting byte of a field within a record, counting from 1.
data_type Provides the data type of the values shown in the field. Valid values are identified in section 5.2.
field_length Provides the length, in bytes, of the field. Does not include field delimiters or "gutter" space between fields.
field_format Provides a printf style string indicating field magnitude and precision. (optional)
unit Indicates the unit of measurement in which the field value is represented. (optional)
scaling_factor Provides a multiplier to be applied to a stored value in a table to recover the original observed value. Original value V_o is calculated from the stored value V_s thus: $V_o = (V_s * scaling\_factor) + value\_offset$. (optional)
value_offset A number to be added to a value stored in a table to recover the original observed value. (See scaling_factor.) (optional)
description Provides a short textual description of the data object. (optional)

4.3 Parsable Byte Stream

Data that do not qualify as either arrays (fixed-length rows of homogeneous elements in *n* dimensions) or fixed width tables (repeating records of scalars) may be classified as either *parsable byte streams* or *encoded byte streams*. The latter will be discussed in the next section.

A parsable byte stream is a stream of bytes, either binary or character, that can be interpreted according to a standard set of rules.

A simple ASCII text file is an example of a parsable byte stream. It consists of a stream of character data, with lines delimited by a standardized set of characters (usually either carriage-return line-feed or just line-feed). Many different applications are able to parse files of this format.

An HTML file is also a parsable byte stream:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
<h1>My First Heading</h1>

<p>My first paragraph.</p>
</body>
</html>
```

The above file could be parsed and displayed by any browser programmed to understand the HTML 4.01 standard.

XML labels, comma separated value (CSV) tables, and SPICE kernels are more examples of parsable byte streams. Additionally, any of several formats of data headers are also recognized by the PDS as parsable byte streams, including: FITS, ISIS, ODL, PDS, TIFF, and VICAR.[2]

The Parsable_Byte_Stream class and its subclasses are used to describe this form of data. The core pieces of information necessary to describe all parsable byte streams are shown below:

```
<Parsable_Byte_Stream>
    <offset unit="byte">0</offset>
    <external_standard_id>PDS3</external_standard_id>
    <external_standard_version_id>3.8</external_standard_version_id>
    <encoding_type>character</encoding_type>
</Parsable_Byte_Stream>
```

offset Provides the starting location, within a file, of the stream of bytes to be parsed; the first byte of the file has an offset of zero.

external_standard_id Indicates the external standard with which the data stream complies; the valid values for this attribute vary depending on the subclass being used, but in general these lists may be found in the (reference to Data Dictionary).

external_standard_version_id If applicable, indicates the relevant version of the external standard (optional).

encoding_type Indicates the encoding of the data, whether “Binary” or “Character”.

Further details on how to populate the Parsable_Byte_Stream subclasses used to describe headers, SPICE kernels, text objects, and XML schemas can be found in the relevant sections of the ap-

[2] Note that this does not imply that all of these formats are necessarily acceptable for archiving observational data.

pendices. However, one parsable byte stream format requires further explanation – the delimiter separated value format (i.e., *delimited tables* or *spreadsheets*).

4.3.1 Delimiter Separated Value Format Description

The delimiter separated value (DSV) (or "spreadsheet") format has been used for some time in a variety of forms for storage and exchange of data between programs and systems.[3] Special cases include the tab separated value (TSV) and comma separated value (CSV) formats. This section describes a general DSV format.

The DSV format is a flexible delimited value format which includes support for TSV and CSV as well as other commonly used delimited value formats. The DSV specification is a generalized version of the format described in RFC 4180.

The DSV format is defined as follows:

1. A file or stream of data may contain one or more records.

2. Each record is separated from adjacent records by a record delimiter, the ASCII carriage-return line-feed pair (denoted `<CR><LF>`), as required for the multipurpose transport of files (RFC 2046).[4] For example:

   ```
   aaa,bbb,ccc<CR><LF>
   zzz,yyy,xxx<CR><LF>
   ```

3. The first record of a file may be an optional header record with the same number of fields as normal records. This header record will contain names corresponding to the fields in the records which follow. For example:

   ```
   field_name1,field_name2,field_name3<CR><LF>
   aaa,bbb,ccc<CR><LF>
   zzz,yyy,xxx<CR><LF>
   ```

4. Within the (optional) header record and each subsequent record, there will be one or more fields; fields are separated by field delimiters, and every record has the same number of fields. All field delimiters are the same. There is no field delimiter after the last field and before the record delimiter. The field delimiter must be one of the following characters: comma (`,`), semi-colon (`;`), vertical bar (`|`), or horizontal tab (`<TAB>`). The first occurrence of one of these characters sets the delimiter to be used for all other fields. For example:

[3] Most of section 4.3.1 is adapted from the white paper by King and Mafi, 2012, with permission.

[4] Note that the current version, 0.3.0.0.a, of the PDS Information Model also permits the use of `<CR>` and `<LF>` individually as record delimiters.

```
aaa,bbb,ccc<CR><LF>
```

defines comma (`,`) as the delimiter and:

```
aaa|bbb|ccc<CR><LF>
aaa|b,b|ccc<CR><LF>
```

defines the vertical bar (—) to be the delimiter. Each record contains three fields. The comma in the second line is part of the second field value.

5. A field may be empty. The interpretation of an empty field will be application and data type dependent. It may be represented as a missing value, not a number (NaN), or an empty string depending on the implicit data type of the field. For example:

```
aaa,bbb,ccc<CR><LF>
aaa,,ccc<CR><LF>
```

has an empty field as the second field in the second record.

6. Leading and trailing spaces in a field are not considered part of the value of a field and should be ignored. For example:

```
aaa,bbb,   ccc<CR><LF>
zzz,yyy,xxxxxx<CR><LF>
```

contains extra spaces in the third field of the first record. These spaces are not part of the value ccc. Note that a field is the content between two delimiters and that a value is the information or data contained within a field. In most cases the field and value are synonymous.

7. Fields may optionally be enclosed in a pair of double quotes. The double quotes must be the first and last characters between delimiters; but the double quotes are not considered part of the field value. Any characters that are enclosed in double quotes are considered literal (including delimiters, and leading or trailing spaces). For example:

```
"aaa,bbb",ccc<CR><LF>
```

the comma in the first field is not treated as a delimiter, but is included in the value. As a result this record consists of two fields (with values of `aaa,bbb` and `ccc`). In another example:

```
aaa,"   bbb",   ccc<CR><LF>
```

while the second and third fields both consist of 6 characters, the spaces in the values of the two fields will differ. The spaces in the second field (which are enclosed in double quotes) are preserved in the value: `   bbb`. The spaces in the third field are ignored in the value: `ccc`.

Double quotes which are not the first and last characters between the delimiters are treated as literal, and part of the field. For example in:

```
aaa,"bbb"b"bbb",ccc"c"ccc<CR><LF>
```

the second and third fields both contain 9 characters with embedded double quotes as part of the values. The enclosing quotes around the second field are not part of the field.

Double quote usage may vary from record to record. There is no requirement that a particular field be quoted in every record.

The following special cases apply to the use of double quotes:

- A field containing two consecutive double quotes (`,"",`) is interpreted as an empty (length 0) field.
- A field which is to be interpreted as containing only a single double quote would be rendered as three consecutive double quotes between the delimiters (`,""",`).

4.3.2 Delimited Tables

The previous section described the format of delimited tables. This section explains how the characteristics of those tables are to be described using the PDS Table_Delimited class.

Figure 4.3 shows the portion of a PDS label that describes a delimited table. A sample of the table contents themselves are shown in Figure 4.4.

When describing delimited table files, the Table_Delimited class should be used only to describe the records of the file containing actual data. Any header records should be described using the Header class, an example of which is provided in Figure **??** (in the appendices, not yet included).

```
<Table_Delimited>
    <offset unit="byte">10489</offset>
    <external_standard_id>PDS CSV</external_standard_id>
    <encoding_type>Character</encoding_type>
    <records>33</records>
    <field_delimiter>comma</field_delimiter>
    <record_delimiter>carriage_return line_feed</record_delimiter>
    <Record_Delimited>
        <fields>385</fields>
        <maximum_record_length unit="byte">3172</maximum_record_length>
        <Field_Delimited>
            <name>SOL</name>
            <field_number>1</field_number>
            <data_type>ASCII_Integer</data_type>
            <description>The Martian solar day during which the
                recorded event occurred.  The day of
                the landing is sol = 1.</description>
        </Field_Delimited>
        <Field_Delimited>
            <name>LST</name>
            <field_number>2</field_number>
            <data_type>ASCII_Integer</data_type>
            <description>The Martian solar day during which the
                recorded event occurred.  The day of
                the landing is sol = 1.</description>
        </Field_Delimited>
        <Field_Delimited>
            <name>SCET</name>
            <field_number>3</field_number>
            <data_type>ASCII_Date_Time_DOY</data_type>
            <description>The time at which the recorded event
                occurred, in UTC format.</description>
        </Field_Delimited>
        <Field_Delimited>
            <name>APID</name>
            <field_number>4</field_number>
            <data_type>ASCII_Integer</data_type>
            <description>Application packet id: the id of the
                telemetry packet queue to which the data was
                directed.  Channel ID=E-1003.  </description>
        </Field_Delimited>
        ...
    </Record_Delimited>
</Table_Delimited>
```

Figure 4.3: Label snippet showing a Table_Delimited class.

```
,, SCET,%12u,%12u,%12u,%12u,%12u,%12u,%12u,%#12x,%12u,%#12x,%12u,%12u,%12u,%12d,%12d,%12u,%12
,, SCET,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,DN,D
,, SCET,E-1003,E-1006,E-1100,E-1101,R-0002,R-0003,R-0004,R-0007,R-0008,R-0009,R-0010,R-0011,R
SOL,LST, SCET ,APID,DATA LENGTH,MSG PKT NUM,CMD SEQ NUM,PKT SEQ NUM,DATA LENGTH,COMMAND CODE,
47 , 11:06:10 , 1997-233T07:31:10.096,,,,,,,,0x           0,,0x           0,          4,
47 , 12:04:36 , 1997-233T08:31:11.457,,,,,,,          23,0x           0,,0x           0,,,,,
47 , 12:04:37 , 1997-233T08:31:12.577,,,,,,,         131,,,0x       0xfff,,,,,           0,,,
47 , 12:25:08 , 1997-233T08:52:17.403,,,,,,,,0x           0,,0x           0,          4,
47 , 13:03:48 , 1997-233T09:32:00.968,,,,,,,         135,,,0x           0,,,,,           0,,,
47 , 13:03:49 , 1997-233T09:32:01.560,,,,,,,         146,,,0x           0,,,,,           0,,,
47 , 13:43:03 , 1997-233T10:12:20.162,,,,,,,,0x           0,,0x           0,          4,
47 , 14:04:21 , 1997-233T10:34:12.704,,,,,,,         135,,,0x           0,,,,,           0,,,
47 , 14:04:22 , 1997-233T10:34:13.368,,,,,,,         131,,,0x           0,,,,,           0,,,
47 , 14:04:22 , 1997-233T10:34:13.952,,,,,,,         146,,,0x           0,,,,,           0,,,
47 , 14:05:09 , 1997-233T10:35:01.973,,,,,,,          23,,,,,,,,           0,,,,,,,,,,,,,,,,,
47 , 14:05:09 , 1997-233T10:35:01.973,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
47 , 14:05:09 , 1997-233T10:35:01.973,,,,,,,,0x           0,,0x           0,,,,,,,,,,,,,,,,,,
47 , 14:05:10 , 1997-233T10:35:03.109,,,,,,,         131,,,0x           0,,,,,           0,,,
47 , 14:05:11 , 1997-233T10:35:03.693,,,,,,,         146,,,0x           0,,,,,           0,,,
47 , 14:05:58 , 1997-233T10:35:51.977,,,,,,,          23,,,,,,,,           0,,,,,,,,,,,,,,,,,
47 , 14:05:58 , 1997-233T10:35:51.977,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
47 , 14:05:58 , 1997-233T10:35:51.977,,,,,,,,0x           0,,0x           0,,,,,,,,,,,,,,,,,,
47 , 14:05:59 , 1997-233T10:35:53.025,,,,,,,         131,,,0x           0,,,,,           0,,,
47 , 14:05:59 , 1997-233T10:35:53.697,,,,,,,         146,,,0x           0,,,,,           0,,,
47 , 14:06:46 , 1997-233T10:36:42.030,,,,,,,          23,,,,,,,,           0,,,,,,,,,,,,,,,,,
47 , 14:06:46 , 1997-233T10:36:42.030,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
47 , 14:06:46 , 1997-233T10:36:42.030,,,,,,,,0x           0,,0x           0,,,,,,,,,,,,,,,,,,
47 , 14:06:47 , 1997-233T10:36:43.174,,,,,,,         131,,,0x           0,,,,,           0,,,
47 , 15:00:58 , 1997-233T11:32:22.913,,,,,,,,0x           0,,0x           0,          4,
47 , 15:04:42 , 1997-233T11:36:12.442,,,,,,,         135,,,0x           0,,,,,           0,,,
47 , 15:04:42 , 1997-233T11:36:13.114,,,,,,,         146,,,0x           0,,,,,           0,,,
47 , 15:48:52 , 1997-233T12:21:35.255,,,,,,,         135,,,0x           0,,,,,           0,,,
47 , 15:48:53 , 1997-233T12:21:35.831,,,,,,,         131,,,0x           0,,,,,           0,,,
47 , 15:48:56 , 1997-233T12:21:38.855,,,,,,,,0x           0,,0x           0,          4,
```

Figure 4.4: Portion of a delimited table file. (The table contains engineering data in the form originally collected from the Mars Pathfinder rover.)

The Table_Delimited class inherits several attributes from the Parsable_Byte_Stream class, and adds several more:

```
<Table_Delimited>
    <offset unit="byte">10489</offset>
    <external_standard_id>PDS CSV</external_standard_id>
    <encoding_type>Character</encoding_type>
    <records>33</records>
    <field_delimiter>comma</field_delimiter>
    <record_delimiter>carriage_return line_feed</record_delimiter>
    <Record_Delimited>
    ...
</Table_Delimited>
```

offset Inherited from the Parsable_Byte_Stream class.
external_standard_id Inherited from the Parsable_Byte_Stream class; must be set to `PDS CSV`.
encoding_type Inherited from the Parsable_Byte_Stream class; must be set to `Character`.
records Indicates the number of data records contained in the data object.
field_delimiter Provides a text description of the character used as a delimiter between fields in each table record. Valid values are `comma`, `horizontal_tab`, `semicolon`, and `vertical_bar`.[5]
record_delimiter Provides a text description of the character or character sequence used as a delimiter between records in a delimiter separated value data object. Valid values are `carriage_return`, `carriage_return line_feed`, and `line_feed`.

Additionally, the Table_Delimited class requires a Record_Delimited class, which is used to describe the structure of every record in the data object.

```
    <Record_Delimited>
        <fields>385</fields>
        <maximum_record_length unit="byte">3172</maximum_record_length>
        <Field_Delimited>
            ...
        </Field_Delimited>
        <Field_Delimited>
            ...
        </Field_Delimited>
        ...
    </Record_Delimited>
```

fields The only required attribute of the Record_Delimited class indicates the number of fields in each record.

[5] (Note, misspelled as `verticle_bar` in the 0.3.0.0.a version of the model.)

maximum_record_length Provides the length, in bytes, of the longest record in the data object, including the record delimiter. (optional)

Since the records in a delimited table are not fixed length, the individual fields may vary in size from one record to the next. However, the number of fields must be the same for all records, and the name and data type of each field must remain the same from line to line. There must be one Field_Delimited class present in the label to describe each field in the table record.

```
<Field_Delimited>
    <name>SOL</name>
    <field_number>1</field_number>
    <data_type>ASCII_Integer</data_type>
    <description>The Martian solar day during which the
        recorded event occurred.  The day of
        the landing is sol = 1.</description>
</Field_Delimited>
```

name Provides a short text string by which a field is known.
field_number Provides the position of a field within a series of fields, counting from 1. (optional)
data_type Provides the data type of the values shown in the field. Valid values are identified in section 5.2.
field_length Provides the length, in bytes, of the field. Does not include field delimiters or "gutter" space between fields. (optional)[6]
field_format Provides a printf style string indicating field magnitude and precision. (optional)
unit Indicates the unit of measurement in which the field value is represented. (optional)
scaling_factor Provides a multiplier to be applied to a stored value in a table to recover the original observed value. Original value V_o is calculated from the stored value V_s thus: $V_o = (V_s * scaling\_factor) + value\_offset$. (optional)
value_offset A number to be added to a value stored in a table to recover the original observed value. (See scaling_factor.) (optional)
description Provides a short textual description of the data object. (optional)

4.4 Encoded Byte Stream

The encoded byte stream structure in the PDS is a byte stream that may only be interpreted after it has been "decoded", according to some well known standard. "Encoded" data may include data that has been compressed and needs to be decompressed before interpretation. Alternatively, it may include data encoded in some form of binary format, such as PDF.

[6]Note: will be replaced with "maximum_field_length" in future versions of the Information Model.

In order to interpret an encoded byte stream, reading software would need to determine the value of the external_standard_id attribute and access the referenced standard for information on how to parse the byte stream. It is PDS policy that only publicly available, open source, widely accepted standards may be used for the encoding of data within the PDS.

There are three subclasses of the Encoded_Byte_Stream class: Encoded_Binary (for files like PDF and MS Word documents), Encoded_Image (for browse, thumbnail, and document images stored in formats like GIF, JPEG, and non-raster TIFF, and Header_Encoded (for binary headers like TIFF headers on raster-formatted TIFF images.

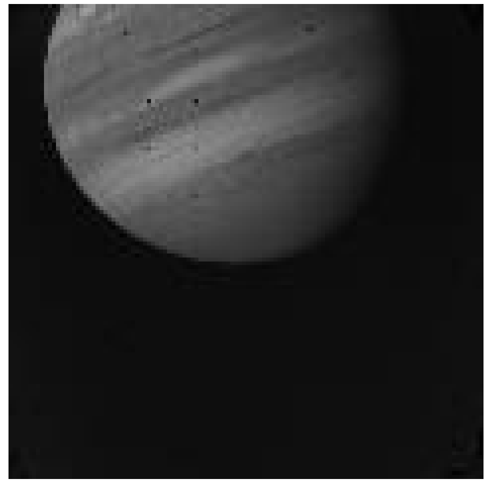


Figure 4.5: Thumbnail image of Jupiter, taken by the Voyager 2 spacecraft. Image is in JPEG format.

```
<Encoded_Image>
    <offset unit="byte">0</offset>
    <external_standard_id>JPEG</external_standard_id>
    <external_standard_version_id>1.02</external_standard_version_id>
    <encoding_type>Binary</encoding_type>
    <description>JPEG-formatted thumbnail version of
        Voyager image 1126J2-029.</description>
</Encoded_Image>
```

offset Provides the starting location, within a file, of the stream of bytes to be parsed; the first byte of the file has an offset of zero.

external_standard_id Indicates the external standard with which the data stream complies; the valid values for this attribute vary depending on the subclass being used, but in general these lists may be found in the (reference to Data Dictionary).

external_standard_version_id If applicable, indicates the relevant version of the external standard (optional).

encoding_type Indicates the encoding of the data, which in this case will always be “Binary”.

description Provides a short textual description of the data object.

Need to fix formatting of above. – EDR

Chapter 5

Data Storage Types

5.1 Attribute Value Types

Attribute value types are used to classify the data types of attribute values used in XML labels. These data types are typically used in class and attribute descriptions in data dictionaries.

Please note that since PDS XML labels contain primarily ASCII (and occasionally UTF-8) characters, the data types in this section do not describe the binary format in which the attribute values are stored, but rather what validation criteria should be applied to the character values. For example, an ASCII_Real value should contain only the characters 0-9 and ".".

5.1.1 Boolean Types

The PDS boolean data type is based on the primitive boolean type as defined in section 3.2.2 of the World Wide Web Consortium (W3C)'s *XML Schema Part 2: Datatypes* (W3C, 2004c).

Data Type	Description	Permitted Values
ASCII_Boolean	True / False indicator	`true` and `false` (lower case only) or `1` (true) and `0` (false)

5.1.2 Date and Time Types

PDS date and time formats are based on the *extended format* of cardinal and ordinal date / time strings as defined in ISO 8601:2004.

The use of a negative sign to indicate dates prior to 1 A.D. is discussed in Section 9.2.

Note that in time components, a decimal fraction may be added to any of the three time elements (hh, mm, or ss) as long as it is the lowest order time element in the representation. This is shown in the table below by `[.fff]`. Currently, only a period "." may be used as the separator between a time element and its fraction. There is no limit to the number of decimal places permitted for the decimal fraction, as long as it is consistent with the precision of the measured value.

Data Type	Description	Permitted Values

ASCII_Date	An ASCII date string in either Day Of Year (DOY) or Year Month Day (YMD) format.	Date value in any of the following forms:
		`[-]yyyy`
		`[-]yyyy-doy`
		`[-]yyyy-mm`
		`[-]yyyy-mm-dd`

ASCII_Date_DOY	An ASCII ordinal date string in DOY format.	Date value in either of the following forms:
		`[-]yyyy`
		`[-]yyyy-doy`

ASCII_Date_YMD	An ASCII calendar date string in YMD format.	Date value in any of the following forms:
		`[-]yyyy`
		`[-]yyyy-mm`
		`[-]yyyy-mm-dd`

ASCII_Date_Time	An ASCII date/time string where the date component may be in either DOY or YMD format.	Date/time value in any of the following forms:[1]
		`[-]yyyy-doyThh[.fff]`
		`[-]yyyy-doyThh:mm[.fff]`
		`[-]yyyy-doyThh:mm:ss[.fff]`
		`[-]yyyy-mm-ddThh[.fff]`
		`[-]yyyy-mm-ddThh:mm[.fff]`
		`[-]yyyy-mm-ddThh:mm:ss[.fff]`
		If Coordinated Universal Time (UTC), `Z` should be appended.

[1]Note: In those rare cases where time information is not available, this format may be truncated down to a date string, including down to just a year.

Data Type	Description	Permitted Values
ASCII_Date_Time_DOY	An ASCII date/time string where the date is in DOY format.	Date/time value in any of the following forms:[1] `[-]yyyy-doyThh[.fff]` `[-]yyyy-doyThh:mm[.fff]` `[-]yyyy-doyThh:mm:ss[.fff]` If UTC, `Z` should be appended.
ASCII_Date_Time_UTC	An ASCII date/time string that must be in Coordinated Universal Time (UTC).	Date/time value of the form: `yyyy-mm-ddThh:mm:ss[.fff]Z`
ASCII_Date_Time_YMD	An ASCII date/time string where the date is in YMD format.	Date/time value in any of the following forms:[1] `[-]yyyy-mm-ddThh[.fff]` `[-]yyyy-mm-ddThh:mm[.fff]` `[-]yyyy-mm-ddThh:mm:ss[.fff]` If UTC, `Z` should be appended.
ASCII_Time	An ASCII time string. May be used for local times on Earth or local solar time on other planets. (See section 9.3.2.4 for a more detailed discussion of local solar times.)	Time value in any of the following forms: `hh[.fff]` `hh:mm[.fff]` `hh:mm:ss[.fff]` If UTC, `Z` should be appended.

[1]Note: In those rare cases where time information is not available, this format may be truncated down to a date string, including down to just a year.

5.1.3 Numeric Types

PDS numeric data types are based on a mixture of primitive and derived types defined in W3C, 2004c and PDS types derived from these. The specific base type for each data type is indicated in the **Description** column in the table below.

Data Types	Description	Permitted Values
ASCII_Integer	An ASCII character representation of a decimal integer. Based on the derived data type xs:integer.	An ASCII string consisting of the digits `0` through `9`, optionally prefixed with a positive "+" or negative sign "−". The value must be in the range `−2147483648` to `2147483647`.

Data Types	Description	Permitted Values
ASCII_NonNegative_Integer	An ASCII character representation of a decimal integer greater than or equal to zero. Based on the derived data type xs:nonNegativeInteger.	An ASCII string consisting of the digits `0` through `9`. The value must be in the range `0` to `4294967295`.
ASCII_Real	An ASCII character representation of a real number. Based on the primitive data type xs:double.	An ASCII string consisting of a mantissa followed, optionally, by the character `E` or `e`, followed by an exponent. The mantissa must be a decimal number, consisting of a sequence of digits `0` through `9` separated by a period "`.`" as a decimal indicator. The value may be optionally prefixed with a positive "+" or negative "–" sign. Leading and trailing zeroes are permitted. If the fractional part is zero, the period and following zeroes may be omitted. The exponent must be an integer. Three special values are permitted: `INF` and `-INF` (positive and negative infinity) and `NaN` (not a number).
ASCII_Numeric_Base2	An ASCII character representation of a number in binary format. This is a PDS defined data type.	An ASCII string consisting of the characters `0` and `1`. Limit 255 characters.
ASCII_Numeric_Base8	An ASCII character representation of a number in octal format. This is a PDS defined data type.	An ASCII string consisting of the characters `0` through `7`. Limit 255 characters.
ASCII_Numeric_Base16	An ASCII character representation of a number in hexadecimal format. Based on the primitive data type xs:hexBinary.	An ASCII string consisting of the characters `0` through `9` and `A` through `F` or `a` through `f`. Limit 255 characters.

Data Types	Description	Permitted Values
ASCII_MD5_Checksum	A 128-bit hash value calculated using the MD5 algorithm (RFC 1321). This is a PDS defined data type.	An ASCII string consisting of the characters 0 through 9 and A through F or a through f. Must be exactly 32 characters in length.

5.1.4 String Types

PDS string data types are based on a mixture of primitive and derived types defined in W3C, 2004c and PDS types derived from these. The specific base type for each data type is indicated in the **Description** column in the table below.

Data Types	Description	Permitted Values
ASCII_AnyURI	A URI or its subclasses URN and URL (See section 6.3.4 for details.) Must conform to syntax described in RFC 3986.[1]	An ASCII string of the form <scheme name>:<hier-part>[?<query>][#<fragment>]
	<scheme name> refers to a specification for assigning identifiers within that scheme. (Examples: file, http, mailto, urn.)	A string consisting of letters, digits, the plus sign "+", hyphen "–", and period ".". The first character must be a letter. Schemes are case-insensitive but the canonical form is lowercase.
	<hier-part> is intended to hold identification information hierarchical in nature.	A string of the form [//<authority>][[/]<path>]

[1]Note: definition differs from schema (see RFA IMG_12).

[2]Unreserved characters include letters, digits, the hyphen "–", period ".", underscore "_", and tilde "~".

[3]Sub-delimiters include the exclamation mark "!", dollar sign "$", ampersand "&", single quote "´", open and close parentheses "(" and ")", asterisk "*", plus sign "+", comma ",", semicolon ";", and equal sign "=".

[4]A percent-encoded character consists of a percent sign "%" followed by two uppercase hexadecimal digits representing the character's numeric value. For example, %20 is the percent-encoded representation of the US-ASCII space character.

[5]Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

<table>
<tr><th>Data Types</th><th>Description</th><th>Permitted Values</th></tr>
<tr><td></td><td><authority> identifies the authority to which the governance of the namespace identified by the remainder of the URI is delegated.</td><td>A optional string of the form [<userinfo>@]<host>[:<port>], terminated by a forward slash "/", question mark "?", number sign "#", or end of the URI.</td></tr>
<tr><td></td><td><userinfo> is an optional component of the <authority>. It consists of a user name and optionally, scheme-specific information about how to gain authorization to access the resource.</td><td>The colon ":" is used as the delimiter between a user name and optional, scheme-specific authorization information. If present, <userinfo>, is followed by an at sign "@" to distinguish it from host info. May include unreserved characters[2] and sub-delimiters.[3]All other characters, including the colon, when used other than as a user name / password delimiter, must be percent-encoded[4].</td></tr>
<tr><td></td><td><host> is a required component of the <authority>. It is most commonly an Internet host, but may also be a registered name not associated with an Internet host.</td><td>Either an IP literal encapsulated in square brackets "[" and "]" or an IPv4 address in dotted-decimal form or a registered name:
"[" <IP-literal> "]"
| <IPv4address> |
<reg-name></td></tr>
<tr><td></td><td><IP-literal> identifies a host by an Internet Protocol literal address, version 6 or later.</td><td>Must begin with a open square bracket "[" and end with a close square bracket "]". Format TBD.</td></tr>
</table>

[1]Note: definition differs from schema (see RFA IMG_12).

[2]Unreserved characters include letters, digits, the hyphen "-", period ".", underscore "_", and tilde "~".

[3]Sub-delimiters include the exclamation mark "!", dollar sign "$", ampersand "&", single quote "'", open and close parentheses "(" and ")", asterisk "*", plus sign "+", comma ",", semicolon ";", and equal sign "=".

[4]A percent-encoded character consists of a percent sign "%" followed by two uppercase hexadecimal digits representing the character's numeric value. For example, %20 is the percent-encoded representation of the US-ASCII space character.

[5]Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

Data Types	Description	Permitted Values
	`<IPv4address>` identifies a host by an Internet Protocol literal address, version 4.	A sequence of four decimal numbers in the range of `0` to `255`, separated by "`.`".
	`<reg-name>` identifies a host by a registered name that is usually intended for lookup within a locally defined host or service name registry. The most common name registry mechanism is the Domain Name System (DNS).	May include unreserved characters[2] and sub-delimiters[3]. All other characters must be percent-encoded[4].
	`<port>` is a port number. It is an optional component of the `<authority>`.	A string of digits delimited from the host by a colon "`:`". If present, the number must be in the range `0` to `65535`.
	`<path>` contains hierarchically organized data, that, combined with the `<query>` component, identifies a resource within the scope of the URI's scheme and naming authority.	The path is terminated by the first question mark "`?`" or number sign "`#`" character, or by the end of the URI. If a URI contains an authority component, then the path component must either be empty or begin with a slash "`/`". If a URI does not contain an authority component, then the path cannot begin with two slash characters "`//`". Not finished.
	`<query>`	

[1] Note: definition differs from schema (see RFA IMG_12).

[2] Unreserved characters include letters, digits, the hyphen "`-`", period "`.`", underscore "`_`", and tilde "`~`".

[3] Sub-delimiters include the exclamation mark "`!`", dollar sign "`$`", ampersand "`&`", single quote "`'`", open and close parentheses "`(`" and "`)`", asterisk "`*`", plus sign "`+`", comma "`,`", semicolon "`;`", and equal sign "`=`".

[4] A percent-encoded character consists of a percent sign "`%`" followed by two uppercase hexadecimal digits representing the character's numeric value. For example, `%20` is the percent-encoded representation of the US-ASCII space character.

[5] Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

Data Types	Description	Permitted Values
	`<fragment>`	
ASCII_DOI	A Digital Object Identifier (DOI). (See section 6.3.3 for details.)	ASCII string of the form: `nn.nnnn/nnn` Limit 255 characters.
ASCII_Identifier	A PDS identifier. (See section 6.2 for details.)	ASCII string beginning with an alphabetic character. Limit 100 characters.
ASCII_LID	A PDS logical identifier. (See section 6.2.1 for details.)	An ASCII string of the form `urn:nasa:pds:xxxx` Limit 255 characters.
ASCII_LIDVID	A PDS versioned identifier (logical identifier plus version id). (See section 6.2.1 for details.)	An ASCII string of the form `urn:nasa:pds:xxxx::M.n`. Limit 255 characters.
ASCII_LIDVID_LID	Either a PDS logical identifier or versioned identifier.	An ASCII string of the form `urn:nasa:pds:xxxx[::M.n]`. Limit 255 characters.
ASCII_VID	A PDS version id. (See section 6.3.1 for details.)	An ASCII string of the form `M.m`, where `M` and `m` are both integers.
ASCII_Directory_Path_Name	A system directory path in UNIX format.	ASCII string of the form: `dir1/dir2/` Limit 255 characters.
ASCII_File_Name	A system file name. (See section 6.5 for details.)	ASCII string of the form: `file_name.file_ext` Limit 255 characters.
ASCII_File_Specification_Name	A system file including directory path, file name, and file extension in UNIX format.	ASCII string of the form: `dir1/dir2/file_name.ext` Limit 255 characters.

[1] Note: definition differs from schema (see RFA IMG_12).

[2] Unreserved characters include letters, digits, the hyphen "–", period ".", underscore "_", and tilde "~".

[3] Sub-delimiters include the exclamation mark "!", dollar sign "$", ampersand "&", single quote "´", open and close parentheses "(" and ")", asterisk "*", plus sign "+", comma ",", semicolon ";", and equal sign "=".

[4] A percent-encoded character consists of a percent sign "%" followed by two uppercase hexadecimal digits representing the character's numeric value. For example, %20 is the percent-encoded representation of the US-ASCII space character.

[5] Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

Data Types	Description	Permitted Values
ASCII_Short_String_Collapsed	An ASCII-encoded text string of limited length with whitespace collapsed. (I.e. multiple spaces, new lines, tabs, and carriage returns are not significant.)	An ASCII string. Limit 255 characters.
ASCII_Short_String_Preserved	An ASCII-encoded text string of limited length with whitespace preserved. (I.e. multiple spaces, new lines, tabs, and carriage returns are significant.)	An ASCII string. Limit 255 characters.
ASCII_Text_Collapsed	An ASCII-encoded text string of unlimited length with whitespace collapsed. (I.e. multiple spaces, new lines, tabs, and carriage returns are not significant.)	An ASCII string.
ASCII_Text_Preserved	An ASCII-encoded text string of unlimited length with whitespace preserved.	An ASCII string.
UTF8_Short_String_Collapsed	A UTF-8 encoded text string of limited length with whitespace collapsed.	A UTF-8 string. Limit 255 bytes.[5]
UTF8_Short_String_Preserved	A UTF-8 encoded text string of limited length with whitespace preserved.	A UTF-8 string. Limit 255 bytes.[5]
UTF8_Text_Preserved	A UTF-8 encoded text string of unlimited length with whitespace preserved.	A UTF-8 string.

[1] Note: definition differs from schema (see RFA IMG_12).

[2] Unreserved characters include letters, digits, the hyphen "–", period ".", underscore "_", and tilde "~".

[3] Sub-delimiters include the exclamation mark "!", dollar sign "$", ampersand "&", single quote "´", open and close parentheses "(" and ")", asterisk "*", plus sign "+", comma ",", semicolon ";", and equal sign "=".

[4] A percent-encoded character consists of a percent sign "%" followed by two uppercase hexadecimal digits representing the character's numeric value. For example, `%20` is the percent-encoded representation of the US-ASCII space character.

[5] Please note that a UTF-8 character string close to 255 *characters* in length may exceed 255 *bytes* in length.

5.1.5 Multi-Valued Data Types

5.1.5.1 Vectors

A vector is a three-component representation of a quantity having both direction and magnitude. It may be used to indicate the position of one point in space relative to another.

Data Type	Description	Permitted Values
Float3Vector[1]	A prototype for a simple vector.	An ASCII string consisting of three comma separated values, where each value is an ASCII real.
Vector[1]	A prototype of a more complex vector.	The three elements of the vector are expressed as three ASCII real values in an associated Vector_Component class.
Vector_Cartesian_3[2]	A generic three element vector defined in Cartesian space. The x, y, and z values provide the magnitude along three orthogonal axes.	Three ASCII real values identified as x, y, and z. Each value may range from -INF to INF.
Vector_Cartesian_3_Acceleration[2]	A three element vector defined in Cartesian space in which the magnitude of the vector indicates the value of the acceleration in the specified direction.	Three ASCII real values identified as x, y, and z. Each value may range from -INF to INF.
Vector_Cartesian_3_Position[2]	A generic three element vector defined in Cartesian space in which the magnitude of the vector indicates the distance of a point in the specified direction from the origin of the reference frame.	Three ASCII real values identified as x, y, and z. Each value may range from -INF to INF.

[1]This data type is obsolete and will be dropped from future versions of the PDS data model. Do not use.

[2]Please note that this data type has not yet been formally defined as of version 0.3.0.0.a of the Information Model Specification. It will appear in the next version of the model and at the present time is considered to be the preferred method for specifying vectors.

Data Type	Description	Permitted Values
Vector_Cartesian_3_Velocity[2]	A generic three element vector defined in Cartesian space in which the magnitude of the vector indicates the value of the velocity in the specified direction.	Three ASCII real values identified as x, y, and z. Each value may range from -INF to INF.

[1]This data type is obsolete and will be dropped from future versions of the PDS data model. Do not use.

[2]Please note that this data type has not yet been formally defined as of version 0.3.0.0.a of the Information Model Specification. It will appear in the next version of the model and at the present time is considered to be the preferred method for specifying vectors.

5.1.5.2 Quaternions

Quaternions have not yet been implemented. – EDR.

A quaternion is a four-component representation of a rotation matrix. This particular definition is focused on the PDS use of quaternions; one should refer to other sources for a more complete discourse on quaternion math.

A quaternion may be used to specify the rotation of one Cartesian reference frame–sometimes referred to as the base frame or the 'From' frame–into coincidence with a second Cartesian reference frame–sometimes referred to as the target reference frame or the 'To' frame. Unlike an Euler rotation where three sequential rotations about primary axes are used, a quaternion rotation is a single action, specified by a Cartesian vector used as the positive axis of the rotation (right hand rule) and the magnitude (an angle) of rotation about that axis.

The quaternion may be thought of as defining the instantaneous orientation–sometimes called 'pointing'–of a structure such as an instrument, antenna, solar array or spacecraft bus, given relative to a specified reference frame (the base frame), at an epoch of interest.

Perhaps of more use is the concept that a quaternion may be used to rotate an arbitrary Cartesian 3-vector defined in one reference frame (e.g. an instrument's reference frame) to an equivalent vector defined in another reference frame (e.g. the frame tied to a spacecraft or the J2000 inertial reference frame).

A quaternion has four components. One of the components is a scalar, a function of the angle of rotation (cosine of half the rotation angle), while the remaining three components are used to specify a vector, given in the base reference frame, about which the rotation will be made. In the PDS context a quaternion has a magnitude of one, and so may be treated as a unit quaternion.

In many cases a time tag (epoch) must be associated with the quaternion because the orientation varies over time. A time tag is not needed if the 'To' and 'From' frames have a fixed offset.

The QUATERNION_DESC element is always to be paired with the QUATERNION element, and will contain a complete description of the formation and rotational sense of the quaternion specified with the QUATERNION keyword, and the structure (organization of the four components) of the quaternion.

In the lingo of the NASA 'SPICE' ancillary information system a rotation matrix is synonymous with a C-matrix–that which may be obtained from a C-kernel. The SPICE Toolkit provides an assortment of routines that deal with quaternions. The SPICE system also provides information about specification of reference frames and time tags suitable for use with quaternions in the SPICE context. The NAIF Node of the PDS can provide additional documentation on quaternions in a spacecraft ancillary data context ('Rotations Required Reading' and 'SPICE Quaternion White Paper').

5.2 Character Data Types

Character data types are used to describe the data formats of character fields in tables.

The values described above in the sections on **Boolean Types**, **Date and Time Types**, and **Numeric Types** may also be used in field descriptions in tables. Table string fields should use the data types describe in the table below.[3]

Data Types	Description	Permitted Values
ASCII_AnyURI	A URI or its subclasses URN and URL (See the description of the ASCII_AnyURI data type in the section above on String Types.)	
ASCII_DOI	A DOI. (See section 6.3.3 for details.)	ASCII string of the form: `nn.nnnn/nnn` Limit 255 characters.
ASCII_File_Name	A system file name. (See section 6.5 for details.)	ASCII string of the form: `file_name.file_ext` Limit 255 characters.

[1] The precise handling of whitespace for these data_types is TBD.

[3] Note that the list of values shown in this table may be significantly reduced in future versions of the Information Model. Please consult your Discipline Node representative before using them.

Data Types	Description	Permitted Values
ASCII_File_Specification_Name	A system file including directory path, file name, and file extension in UNIX format.	ASCII string of the form: `dir1/dir2/file_name.ext` Limit 255 characters.
ASCII_Identifier	A PDS identifier. (See section 6.2 for details.)	ASCII string beginning with an alphabetic character. Limit 100 characters.
ASCII_LID	A PDS logical identifier. (See section 6.2.1 for details.)	An ASCII string of the form `urn:nasa:pds:xxxx` Limit 255 characters.
ASCII_LIDVID	A PDS versioned identifier (logical identifier plus version id). (See section 6.2.1 for details.)	An ASCII string of the form `urn:nasa:pds:xxxx::M.n`. Limit 255 characters.
ASCII_VID	A PDS version id. (See section 6.3.1 for details.)	An ASCII string of the form `M.m`, where `M` and `m` are both integers.
ASCII_String	An ASCII-encoded text string. The length is set with the field_length attribute.[1]	An ASCII string.
UTF8_String	A UTF-8 encoded text string. The length is set with the field_length attribute.[1]	A UTF-8 string.

[1] The precise handling of whitespace for these data_types is TBD.

5.3 Binary Data Types

Binary data types are used to describe data formats of fields in binary tables and array element formats in arrays.

5.3.1 Integers

5.3.1.1 Signed LSB Integers

This section describes signed integers stored in Least Significant Byte (LSB) first (also known as *little-endian*) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3, up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

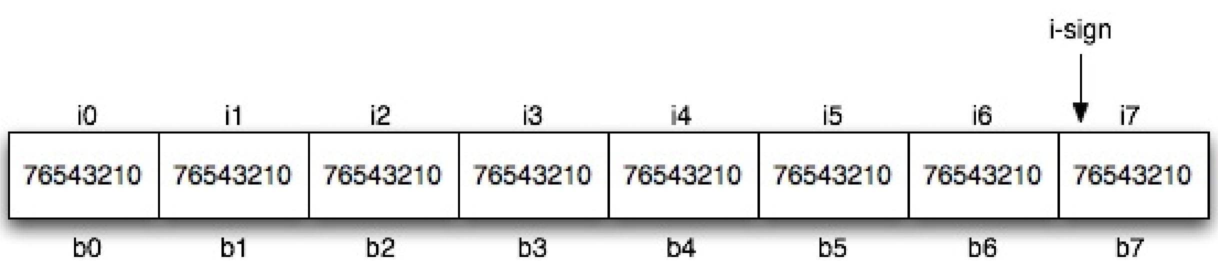
8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

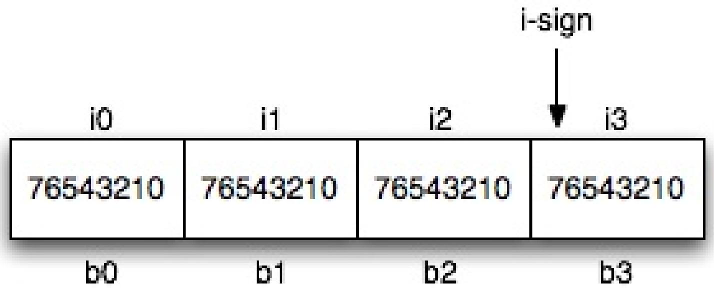
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

All negative values are represented in two's complement.

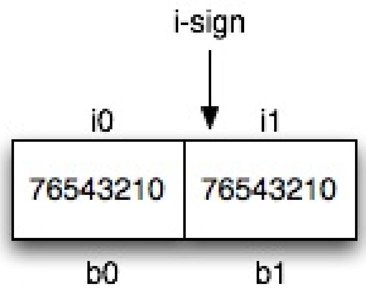
SignedLSB8



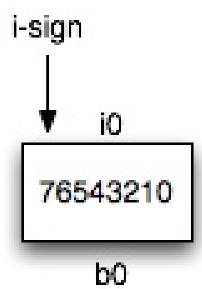
SignedLSB4



SignedLSB2



SignedByte



5.3.1.2 Unsigned LSB Integers

This section describes unsigned integers stored in LSB format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when reading a file (e.g., read byte b0 first, then b1, b2 and b3)

i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7), in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

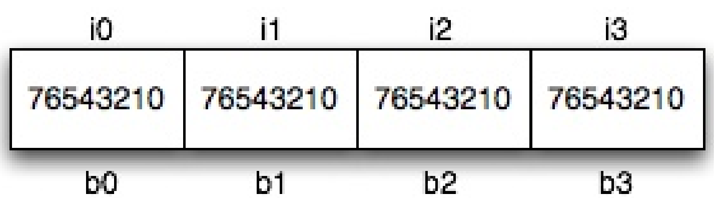
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

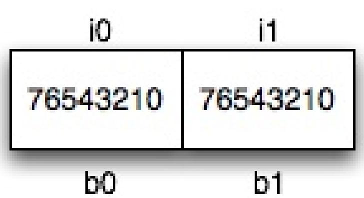
UnsignedLSB8

TBD

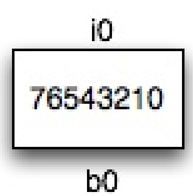
UnsignedLSB4



UnsignedLSB2



Unsigned Byte



5.3.1.3 Signed MSB Integers

This section describes the signed integers stored in Most Significant Byte (MSB) first (also known as *big-endian*) order. In this section the following definitions apply:

b0 – b7 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3 up through b7)

i-sign Integer sign bit – bit 7 in the highest order byte

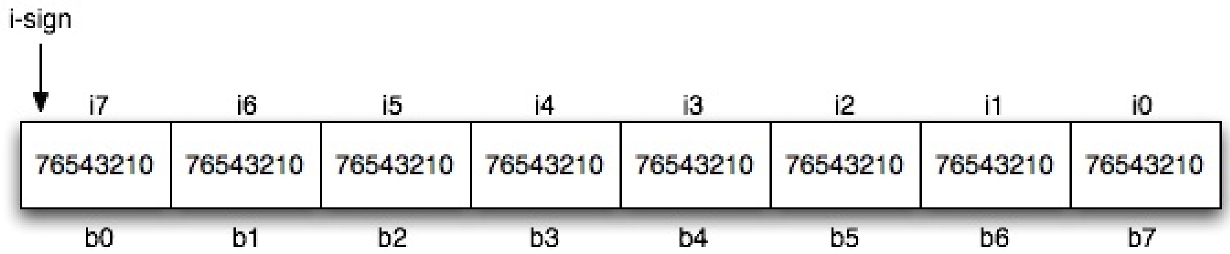
i0 – i7 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

8-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}
In i4, bits 0-7 represent 2^{32} through 2^{39}
In i5, bits 0-7 represent 2^{40} through 2^{47}
In i6, bits 0-7 represent 2^{48} through 2^{55}
In i7, bits 0-6 represent 2^{56} through 2^{62}

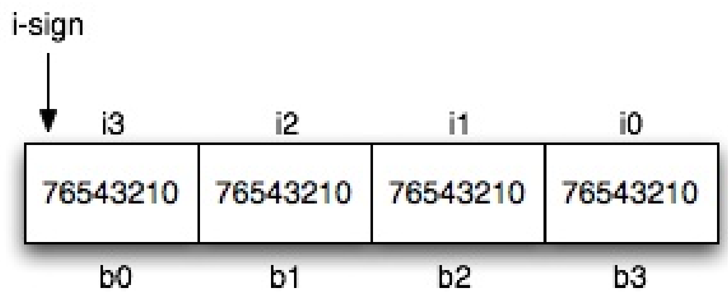
4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-6 represent 2^{24} through 2^{30}

2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-6 represent 2^8 through 2^{14}

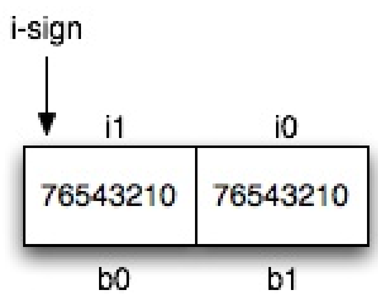
SignedMSB8



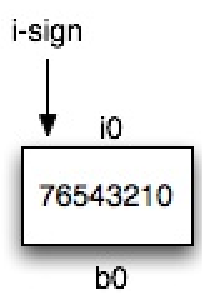
SignedMSB4



SignedMSB2



SignedByte



5.3.1.4 Unsigned MSB Integers

This section describes unsigned integers stored in MSB format. In this section the following definitions apply:

b0 – b3 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, and b3)

i0 – i3 Arrangement of bytes in the integer, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = bit 0, highest value = bit 7) in the following way:

4-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}
In i2, bits 0-7 represent 2^{16} through 2^{23}
In i3, bits 0-7 represent 2^{24} through 2^{31}

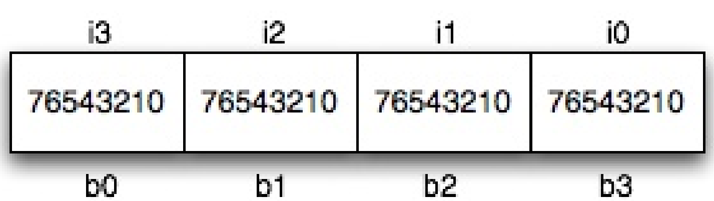
2-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7
In i1, bits 0-7 represent 2^8 through 2^{15}

1-byte integers:
In i0, bits 0-7 represent 2^0 through 2^7

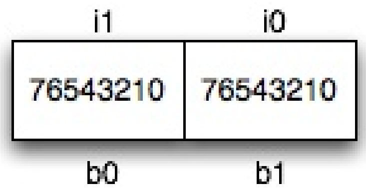
UnsignedMSB8

TBD

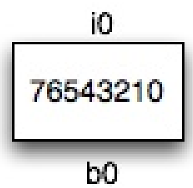
UnsignedMSB4



UnsignedMSB2



UnsignedByte



5.3.2 Reals

This section describes the internal format of IEEE-format floating-point numbers. In this section the following definitions apply:

Need to convert the following to MSB and LSB real values: IEEE754LSBDouble, IEEE754LSBSingle, IEEE754MSBDouble, and IEEE754MSBSingle.

b0 – b9 Arrangement of bytes as they appear when read from a file (e.g., read b0 first, then b1, b2, b3, etc.)

m-sign Mantissa sign bit

int-bit In 10-byte real format only, the integer part of the mantissa, assumed to be "1" in other formats, is explicitly indicated by this bit

e0 – e1 Arrangement of the portions of the bytes that make up the exponent, from lowest order to highest order. The bits within each byte are interpreted from right to left (e.g., lowest value = rightmost bit in the exponent part of the byte, highest value = leftmost bit in the exponent part of the byte) in the following way:

8-bytes (double precision):
In e0, bits 4-7 represent 2^0 through 2^3
In e1, bits 0-6 represent 2^4 through 2^{10}

Exponent bias = 1023

4-bytes (single precision):
In e0, bit 7 represents 2^0
In e1, bits 0-6 represent 2^1 through 2^7

Exponent bias = 127

m0 – m7 Arrangement of the portions of the bytes that make up the mantissa, from highest order fractions to the lowest order fraction. The order of the bits within each byte progresses from left to right, with each bit representing a fractional power of two, in the following way:

8-bytes (double precision):
In m0, bits 3-0 represent $1/2^1$ through $1/2^4$
In m1, bits 7-0 represent $1/2^5$ through $1/2^{12}$
In m2, bits 7-0 represent $1/2^{13}$ through $1/2^{20}$
In m3, bits 7-0 represent $1/2^{21}$ through $1/2^{28}$
In m4, bits 7-0 represent $1/2^{29}$ through $1/2^{36}$
In m5, bits 7-0 represent $1/2^{37}$ through $1/2^{44}$
In m6, bits 7-0 represent $1/2^{45}$ through $1/2^{52}$

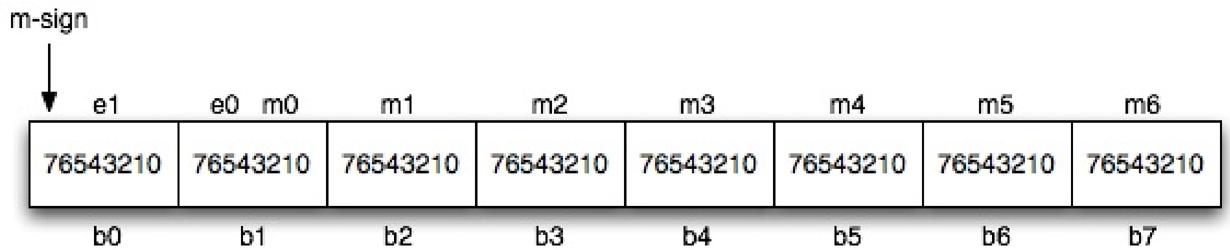
4-bytes (single precision):
In m0, bits 6-0 represent $1/2^1$ through $1/2^7$
In m1, bits 7-0 represent $1/2^8$ through $1/2^{15}$
In m2, bits 7-0 represent $1/2^{16}$ through $1/2^{23}$

The following representations all follow this format:

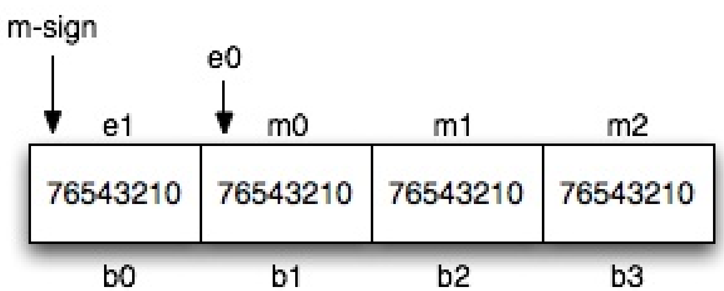
$$1.\mathit{mantissa} \text{ x } 2^{(exponent-bias)}$$

Note that the integer part ("1.") is implicit in all formats as described above. In all cases the exponent is stored as an unsigned, biased integer (that is, the stored exponent value - bias value = true exponent).

IEEE754Double



IEEE754Single

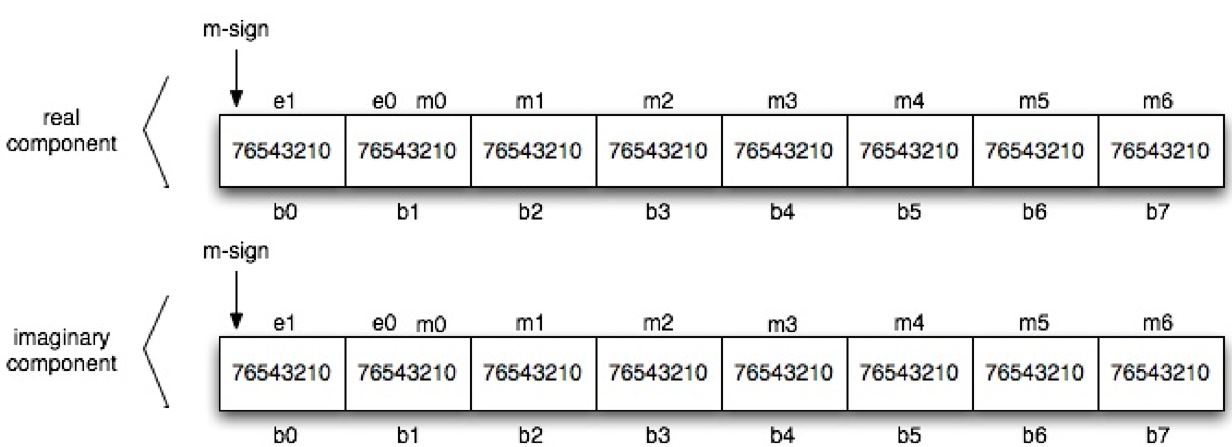


5.3.3 Complex

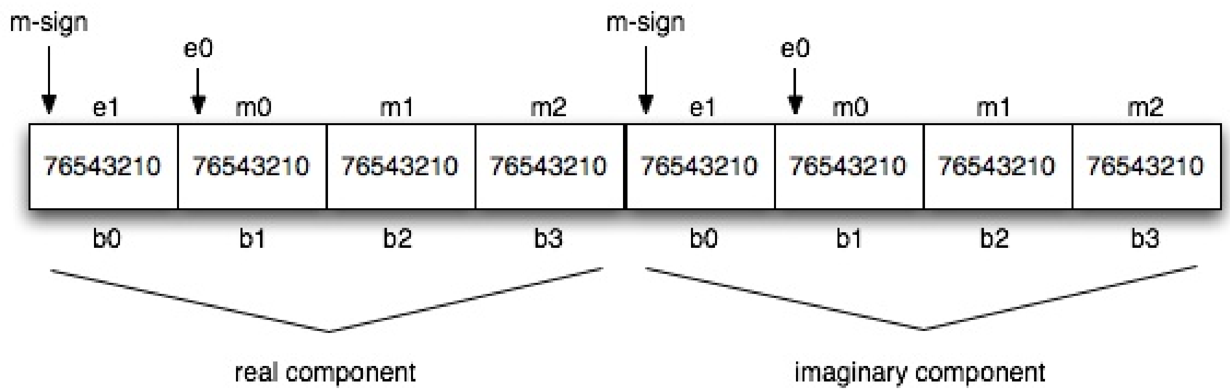
IEEE complex numbers consist of two IEEE_REAL format numbers of the same precision, contiguous in memory. The first number represents the real part and the second the imaginary part of the complex value.

Need to convert the following to MSB and LSB complex values: ComplexLSB16, ComplexLSB8, ComplexMSB16, and ComplexMSB8.

ComplexB16



ComplexB8



5.3.4 Bit Strings

SignedBitString

TBD

UnsignedBitString

TBD

Chapter 6

Nomenclature Rules

Make sure to specify permitted character sets!

6.1 Namespaces

In order to avoid collisions among namespaces in the PDS, namespaces used in PDS4 labels must be chosen from among a predefined set or created using established formation rules. Table 6.1 shows the existing namespaces and their abbreviations that have been defined by the PDS.

Mission namespace abbreviations and identifiers are also set by the PDS. Consult your PDS discipline node to determine the correct namespace to use for your mission. In general, mission namespace identifiers shall conform to the following pattern:

```
http://pds.nasa.gov/pds4/mission/<mission abbreviation>/v<version#>
```

Any instrument (or other) team that requires a namespace distinct from its mission's namespace shall build its namespace upon the mission namespace:

```
http://pds.nasa.gov/pds4/mission/<mission abbr.>/<inst or team abbr.>/v<version#>
```

If a separate namespace is necessary for describing data from an investigation that is not mission related, consult your discipline node for assistance in determining how to formulate it.

Namespaces from multiple PDS disciplines, missions, and other investigations may be referenced in PDS products.

Abbr.	Namespace Identifier	Namespace Entity
xs	http://www.w3.org/2001/XMLSchema	XML Schema
xsi	http://www.w3.org/2001/XMLSchema-instance	XML Schema-Instance
sch	http://purl.oclc.org/dsdl/schematron	Schematron schema
pds	http://pds.nasa.gov/pds4/pds/v03	PDS4 common
atm	http://pds.nasa.gov/pds4/atm/v1	Atmospheres Node
geo	http://pds.nasa.gov/pds4/geo/v1	Geosciences Node
img	http://pds.nasa.gov/pds4/img/v1	Imaging Node
naif	http://pds.nasa.gov/pds4/naif/v1	Navigation and Ancillary Information Facility
ppi	http://pds.nasa.gov/pds4/ppi/v1	Planetary Plasma Interactions Node
rings	http://pds.nasa.gov/pds4/rings/v1	Planetary Rings Node
rs	http://pds.nasa.gov/pds4/rs/v1	Radio Sciences
sbn	http://pds.nasa.gov/pds4/sbn/v1	Small Bodies Node
calib	http://pds.nasa.gov/pds4/calib/v1	Calibration Discipline
cart	http://pds.nasa.gov/pds4/cart/v1	Cartography Discipline
geom	http://pds.nasa.gov/pds4/geom/v1	Geometry Discipline

Table 6.1: Pre-defined namespaces for use in PDS4 labels.

6.2 Product Identifiers

Every product in the PDS4 registry must have a globally unique identifier. In the PDS, we refer to this as a *versioned identifier* or *LIDVID*. These identifiers take the form of a Uniform Resource Name (URN), and must therefore comply with the syntax described in RFC 2141.

PDS products must have LIDVIDs that begin with the sequence of characters "`urn:nasa:pds:`". LIDVIDs must contain only ASCII characters and are limited to 255 characters in length.

Note that a URN is a subset of Uniform Resource Identifier (URI) that uses the *urn* scheme; it carries no implication as to the availability of an identified resource.

6.2.1 Logical Identifiers vs. Versioned Identifiers

PDS LIDVIDs, minus the version information, are referred to as *logical identifiers* or *LIDs*. The logical identifier is used to denote all versions of a product collectively.

```
LIDVID = LID``:'' version_id
```

PDS LIDs are constructed by assembling unique identifiers for each of the constituent components of an archive hierarchy that are relevant to the particular product. Thus, a bundle LID is composed of `urn:nasa:pds:` plus a unique bundle identifier. A collection LID appends a unique collection identifier onto a bundle LID, and a basic product LID appends a unique product identifier onto a collection LID:

```
urn:nasa:pds:<bundle id>
urn:nasa:pds:<bundle id>:<collection id>
urn:nasa:pds:<bundle id>:<collection id>:<basic product id>
```

Note that the colon character (`:`) shall be used exclusively to separate the `urn` scheme identifier, the National Aeronautics and Space Administration (NASA) and PDS components, and the bundle, collection, and basic product components. It shall not be used to separate sub-components within the bundle, collection, and basic product identifiers.

Applications which provide access to PDS products identified by LID are expected to return the most recent version of the product. They may additionally provide access to earlier versions of the product.

6.2.2 Bundle Identifiers

Bundle LIDs must be unique across the PDS. Therefore, a bundle identifier shall be constructed in such a way as to ensure the uniqueness of all of the bundle's constituent collections and basic products, while not monopolizing a generic term that precludes related bundles from using appropriate names.

For example, a bundle which includes only raw data, documentation, and geometry information related to one of many instruments on a spacecraft, shall not use only a generic identifier like a mission abbreviation in forming a bundle LID:

```
incorrect: urn:nasa:pds:mpf
```

Rather, the identifier should be as specific as is possible while being common across all of the component collections of the bundle:

```
correct: urn:nasa:pds:mars-mpf-imp-raw
```

The unique identifier used in constructing a bundle LID is not designed to be a human-friendly text string; rather it is designed to ensure global uniqueness. Therefore, components of the identifier should themselves be short, using abbreviations where sensible. Components derived from targets, missions, instruments, spacecraft, telescopes, data reduction levels, and investigators may all be used.

In general, the left most component of bundle identifiers shall be named for the primary organizing principle for the data. In other words, if the bundle contains only data from a particular mission, the mission shall be the first component of the identifier. If the bundle contains only context data, ”context” shall be the first component of the identifier. If the bundle contains only data from a particular observation campaign (like the SL-9 impact), an identifier for that campaign shall be the first component of the identifier.

The unique identifier shall be constructed with the most generic component first and the least generic component last. From the above example:

- Within the set of all data acquired at Mars, this bundle contains only data acquired by the Mars Pathfinder (MPF) spacecraft.
- Within the set of all data acquired by the MPF spacecraft, this bundle contains only data acquired by the Imager for Mars Pathfinder (IMP) instrument.
- Within the set of all data acquired by the IMP instrument, this bundle includes only raw data.

This construction permits the naming of related bundles for derived data and data acquired by other instruments.

bundle logical identifier (lid) = “urn:nasa:pds:” <specific identifier>
bundle versioned identifier (lidvid) = <bundle logical identifier> “::” <bundle version id>

When constructing the specific identifier component of the bundle identifier, underscores shall be used within terms, and dashes shall be used between terms. (Note: based on this, need to fix other examples throughout this document. – EDR) Thus, an underscore carries the connotation of “and”.

Since logical identifiers are frequently used by data preparers as file specification names or components thereof, the forward slash character (/) is not permitted within a logical id.

LIDs shall be constructed using all lowercase characters.

6.2.3 Collection Identifiers

Collection identifiers are constructed by appending a colon (:), then a specific identifier onto the bundle LID.

collection logical identifier (LID) = <bundle logical identifier> “:” <specific identifier>
collection versioned identifier (LIDVID) = <collection logical id> “::” <collection version id>

As for the bundle specific identifier, the collection specific identifier shall be constructed on the principal of most generic component first and least generic component last. Collection specific identifiers shall also include the collection type as one of the components. Given that bundles may be organized around vastly varying concepts, the construction of the specific identifier component of a collection may vary greatly from one bundle to another.

```
urn:nasa:pds:mars-mpf-asimet:data-raw
urn:nasa:pds:mars-mpf-asimet:data-reduced
urn:nasa:pds:mars-mpf-imp-raw:data
urn:nasa:pds:mars-mpf-imp-raw:document
urn:nasa:pds:mars-mpf-imp-raw:xml-schema
urn:nasa:pds:mars-mfex:calibration-rvrcam
urn:nasa:pds:mars-mfex:document
urn:nasa:pds:mars-mfex:data-apxs-raw
urn:nasa:pds:mars-mfex:data-apxs-derived
urn:nasa:pds:mars-mfex:data-rvrcam-raw
urn:nasa:pds:mars-mfex:data-rvreng-raw
urn:nasa:pds:mars-mfex:data-rvreng-reduced
```

6.2.4 Basic Product Identifiers

Basic product identifiers are formed by appending a colon (`:`) and a specific product identifier to the collection LID in which the basic product is a primary member.

product logical identifier (lid) = <collection logical identifier> ":" <specific identifier>
product versioned identifier (lidvid) = <product logical identifier> "::" <product version id>

The formation of the specific identifier component of basic product LIDs is up to the data provider in consultation with their PDS discipline node.

Many data providers choose to use the file name for specific identifier. In this case, it shall consist only of the file name base of the label file (i.e., without the file name extension). If the file name base is not unique within the collection, information (such as the relative directory path) shall be pre-pended to it to make it unique.

Rules:

- Must be unique across the PDS.
- (character set is same as for file names)

6.2.5 Context Product Identifiers

The primary version of every context product is archived with the PDS Engineering Node as part of a PDS-wide context bundle. Thus, the rules for generating context product LIDs are more tightly constrained than those for other products.

The LID for the PDS context bundle is:

```
urn:nasa:pds:context
```

The context collections within this bundle are organized primarily by the type of context product. Therefore, the collection LIDs take the form of one of the following:

```
urn:nasa:pds:context:agency
urn:nasa:pds:context:instrument_host
urn:nasa:pds:context:instrument
urn:nasa:pds:context:investigation
urn:nasa:pds:context:node
urn:nasa:pds:context:other
urn:nasa:pds:context:personnel
urn:nasa:pds:context:resource
urn:nasa:pds:context:target
urn:nasa:pds:context:telescope
```

Note that several of these collections contain multiple sub-types of context products. For example, the **personnel** collection will contain both PDS_Affiliate and PDS_Guest context products. Similarly, context products of type Facility will be included in the **instrument_host** collection.

The individual context product LIDs are formed by appending unique identifiers to the collection LIDs. As with the formation of other types of LIDs, the components used in context LID formation should be abbreviated where sensible.

```
urn:nasa:pds:context:instrument_host:co
urn:nasa:pds:context:instrument_host:vgr1
urn:nasa:pds:context:investigation:dspse
urn:nasa:pds:context:investigation:hst
urn:nasa:pds:context:investigation:mgn
urn:nasa:pds:context:node:img
```

Note that because there is frequently overlap in the acronyms used for instruments on various spacecraft, the instrument unique identifier shall be formed by appending an abbreviation for the instrument host (or other uniquely identifying information) onto the end of the instrument.

```
urn:nasa:pds:context:instrument:iss_co
```

```
urn:nasa:pds:context:instrument:iss_mr9
```

Similarly, unique identifiers formed from people's names should consist of their last name, followed by their first initial, and as many additional characters or initials from their name as are necessary to make the identifier unique.

```
urn:nasa:pds:context_bundle:pds-affiliate:arvidson_l
urn:nasa:pds:context:pds-affiliate:arvidson_r
urn:nasa:pds:context:pds-guest:runkle_a
```

In contrast to most of the above, when forming LIDs for targets, the full name of the target shall be used. In order to ensure uniqueness in target identifiers, the target name shall be followed by the target type:

```
urn:nasa:pds:context:target:jupiter_planet
urn:nasa:pds:context:target:metis_asteroid
urn:nasa:pds:context:target:metis_satellite
urn:nasa:pds:context:target:s-rings_ring
urn:nasa:pds:context:target:sl9_comet
urn:nasa:pds:context:target:spica_star
urn:nasa:pds:context:target:stim-lamp_calibration
urn:nasa:pds:context:target:sun_sun
```

Uncertain how to make convention for naming of resources. – EDR

```
urn:nasa:pds:context:resource:?
```

6.3 Other Identifying Information

6.3.1 Version IDs

Version IDs are used for all types of products, including basic products, collections, and bundles.

Note that in all cases, the incrementing of a version identifier implies that a user of that product should use the latest version of the product. In other words, incrementing a version indicates that previous versions of the product have been superseded. (It is, of course, possible that a user may, in specific cases, deliberately choose to use older versions of a product. As long as those products were archived with the PDS, they will still be available, albeit potentially offline.)

Formation Rules:

- Version identifiers are appended to logical identifiers to form versioned identifiers. They are separated from the logical identifier by a double colon ("::").
- Version IDs must be of the form M.m where "M" and "m" are both integers. "M" is the "major" component of the version and "m" is the "minor" component of the version.
- The major number is initialized to one for archive products. (Zero may be used for sample products or test run products that are not intended for the archive.) The minor number is initialized to zero.
- Whenever the major number is incremented, the minor number is re-set to zero.
- The minor portion of the version is **not** pre-padded with zeros; it is simply incremented as an integer. Thus, "1.1" and "1.10" are different versions; "1.01" is invalid.

It is up to the determination of each PDS discipline node, working in consultation with data preparers, to determine the criteria for how and under what circumstances version numbers will be incremented for a particular archive. Specific versioning conventions for every archive must be detailed in the respective product documentation. Note that depending upon the criteria that a given node/mission combination selects, it is possible in some cases that higher versioned products may be created without their lower versioned counterparts existing.

6.3.2 Local Identifiers

Change to "Local IDs"?

Local identifiers are only required to be unique within an individual product label. They are primarily used for navigating among the different portions of the label and can be used, for example, to tie together a data object with an Object_Statistics class describing it, to to make clear the relationship between two Image_Grayscale classes (a primary and a browse) and the one or two File classes identifying the data they describe.

The construction of these identifiers is therefore largely up to the data provider, with the exception of the character set restrictions. There is also one recommendation, to make these identifiers simpler for humans to find in a label:

Rules:

- Must be unique within the containing label.
- The character set for local identifiers is restricted to ASCII letters, number, the underscore, the colon, the dash, and the space. Numeric codes for these ASCII characters to be added. Period? Forward slash? Semi-colon? - No space

formation rule: use class name with incrementing number appended, underscore, prepended with zeroes so that all local identifiers of the same class type have the same number of digits (This is to preserve the ability to do alphabetical sorting.)

6.3.3 Digital Object Identifiers

TBD – Probably need to add a description of these. – EDR

6.3.4 Uniform Resource Identifiers

TBD – Probably need to add a description of these. – EDR

6.3.5 Names

TBD - Is this still needed?

Rules:

-

Recommendations:

-

6.3.6 Titles and Alternate Titles

Titles and alternate titles are intended for human consumption. Their primary use is to be displayed alongside a product when it is shown in some sort of product browser. It is not required to be unique. The specificity of the title is entirely up to the data provider, and is constrained only the manpower and resources necessary to populate it. Both of the following are acceptable for titles:

”LORRI image of Io; lat: 4 degS, lon: 164 degW; time: 2007-03-01T00:35 UTC”

”This image of Io, captured by New Horizon’s Long Range Reconnaissance Imager, captured the active Tvashtar plume. The image was acquired on March 1, 2007.”

The first of these two titles could be generated automatically by data production pipeline software. The second required post-production modification to incorporate observed feature information. Either is acceptable.

UTF-8 printable characters (explicitly disallowing roughly 30 ASCII characters non-printable characters)

limit to 255 **bytes** in length (up to data providers to be careful if using UTF-8 to verify byte length) (character type definition?)

Rules:

- Titles must be no more than 255 characters in length.
- Titles are text strings. The character set is restricted to the US-ASCII? character set.

Recommendations:

- None.

How are alternate titles used?

6.4 Directories

Reserved directory names: browse, calibration, context, data, document, geometry, supplemental (or about), xml_schema

same character set as filenames except no periods

Rules:

-

Recommendations:

-

Although NTFS allows each path component (directory or filename) to be 255 characters long and paths up to about 32767 characters long, the Windows kernel only supports paths up to 259 characters long if no UNC is used for addressing. – Need to follow up on this. See the section on “Maximum Path Length Limitation” on the following web page: http://msdn.microsoft.com/en-us/library/aa365247(VS.85).aspx

6.5 Filenames

Although modern operating systems are extremely permissive when it comes to filenames, best practices argue against taking full advantage of this. For example, many characters that are legal in filenames require escaping and can create numerous problems when moving files from one operating system to another. Similarly, 255 character filenames, while permitted, are often inconvenient to users who can’t distinguish between similarly named lengthy filenames in many views on various operating systems.

Rules:

- Filenames must be unique within directories.
- Filenames must be no longer than 255 characters (although see path length restrictions discussed in the section on directory names).
- Filenames must be case-insensitive. (In other words, “MyFile.txt” and “myfile.txt” are not permitted in the same directory.)
- The character set is restricted to A-Z (ASCII 0x41 through 0x5A), a-z (ASCII 0x61 through 0x7A), 0-9 (ASCII 0x30 through 0x39), dash ”-” (ASCII 0x2D), underscore ”_” (ASCII 0x5F), and period ”.” (ASCII 0x2E). (we may re-consider this later)
- Filenames may not begin or end with a dash, underscore, period, or space.
- Filenames must begin with an alphanumeric character (A-Z or a-z or 0-9).

Recommendations:

- Don’t abuse the permitted 255 character length for filenames. Try to keep them no longer than 30 to 40 characters in general. **Archives can fail peer review for being unnecessarily difficult to use as much as for violating actual PDS standards!**

Still have to discuss file extensions.

filenames must have at least one period followed by an extension. They may have more than one period, but PDS will consider all periods other than the final one to be part of the basename. (file base + file extension) remember all SPICE file extensions need to be reserved

Need to include version information in filenames.

(See: “Naming Files, Paths, and Namespaces” from MSDN and Wikipedia topic “Filename”.)

6.5.1 Prohibited Filenames

The following filenames have specific purposes on some operating systems; therefore, their use within PDS archives is prohibited.

a.out
core

6.5.2 Reserved Filenames

bundle.xml
catalog.xml
readme.html
readme.txt

6.5.3 Prohibited Basenames

Because the following basenames have specific purposes on some operating systems, their use within PDS archives is prohibited.

AUX	COM1	COM2	COM3	COM4	COM5	COM6	COM7
COM8	COM9	CON	LPT1	LPT2	LPT3	LPT4	LPT5
	LPT6	LPT7	LPT8	LPT9	NUL	PRN	

6.5.4 Reserved Basename Components

bro	browse
cal	calibration
col	collection
con	context
dat	data
doc	document
geo	geometry
inh	instrument_host
ins	instrument
int	investigation
inv	inventory
mis	miscellaneous
nod	node
res	resource
sch	schema
spi	spice_kernels
tar	target

6.5.5 Reserved File Extensions

The file extensions listed in Table 6.2 are reserved. A brief description for each is provided in the table.

Note that the presence of any given file extension in this table should in no way be construed as to imply that the associated format is acceptable for data archiving purposes. Please consult your Discipline Node for assistance in determining acceptable formats for that purpose.

6.6 Classes

The following rules and recommendations for the naming of classes in the PDS data model apply to all levels of the model: common, node, and mission.

Rules:

- The character set is restricted to ASCII letters, numbers, and the underscore. The first character of the class name must be a letter.

Extension	Description
.bc	SPICE binary CK file (spacecraft and instrument orientation data)
.bdb	SPICE binary DBK file (databases in SPICE format)
.bds	SPICE binary DSK file (digital shape data for natural bodies)
.bep	SPICE binary EK file, Science Plan component (events information)
.bes	SPICE binary EK file, Sequence component (events information)
.bpc	SPICE binary PCK file (high-accuracy natural body rotation data)
.bsp	SPICE binary SPK file (trajectory and ephemeris data)
.csv	delimited tabular data file (comma separated value)
.dat	generic binary data file (can be used for files containing different types of digital objects)
.doc or .docx	Microsoft Word® file
.fit or .fts	Flexible Image Transport System file
.gif	Graphics Interchange Format file
.hdr	generic header file, usually character data
.htm or .html	HyperText Markup Language formatted file
.img	raster formatted image data (may contain headers)
.jp2	JPEG 2000 (JP2) formatted file
.jpg or .jpeg	Joint Photographic Experts Group formatted file
.lbl	PDS3 or other format label
.nrb	SPICE text orbit number file, ascending or descending node numbering (orbit start and stop times)
.orb	SPICE text orbit number file, periapsis or apoapsis numbering (orbit start and stop times)
.pdf	Portable Document Format
.qub or .qut	multispectral qube data file
.sch	Schematron file
.tab	fixed-width tabular data file
.ten	SPICE text EK file, Experimenter's Notebook component (events information)
.tf	SPICE text FK file (reference frames definitions)
.ti	SPICE text IK file (instrument parameters and FOV definitions)
.tif or .tiff	Tagged Image Format File
.tls	SPICE text LSK file (leapsecond information)
.tm	SPICE text MK file (meta-kernels listing kernels to be used together)
.tpc	SPICE text PCK file (natural body rotation and size/shape constants)
.tsc	SPICE text SCLK file (spacecraft clock correlation data)
.txt	plain text file
.vic	Video Image Communication and Retrieval file
.xml	Extensible Markup Language formatted file
.xsd	XML Schema file
.zip	Zip compressed archive file

Table 6.2: Reserved File Extensions

- Each component of the class name shall begin with an uppercase character; all other characters shall be lowercase, except when the class name incorporates an acronym. (Ex. Stream_Delimited_Field, SPICE_Kernel, Array_2D)
- The class name must not exceed 255 characters.
- The word order for the components of the class name shall be most significant first (Ex. Product_Table_Character, rather than Character_Table_Product).

Recommendations:

- Whenever possible, when creating subclasses of existing classes, use portions of the parent class name in the child class name to make the relationship between the classes apparent. (Ex. Table_Base, Table_Base_Character or Image_3D, Image_Color, Image_Color_Anaglyph)
- Use widely recognized and accepted terms for class name components, that clearly indicate the nature of a particular class. (Examples to avoid: Image_PanMos_Proj_JMR, Fpu_Bin_Table, Comp_12_8_Parm)

Camel Case - MSB (or do what make sense) follows rules for sub-classes

6.7 Attributes

Rules:

-

Recommendations:

-

rightmost component must correspond to an existing data element concept no articles

tool that provides pull down list of data element concepts for user to select from,

lower case - LSB

rules for common and node dictionaries but not for mission?

6.8 Attribute Values

Rules:

-

enumerated values - case no restrictions, but must match enumerated values (i.e. no limitation on person who created dictionary, but limitation on data preparer is that they must follow DD), UTF-8

Recommendations:

-

Part II

Data Content Standards

Chapter 7

Context Documentation

Scientific data cannot be properly understood and interpreted without knowledge of the context within which the data were acquired. The PDS requires that contextual information for all archived data be included within the archive or be identified and readily available from external publishers.

Contextual information includes descriptions of such varied subjects as the instrument that acquired the data, the mission, spacecraft or other facility from which the instrument operates, and the target of the investigation. Additional documentation should be provided on the key personnel associated with an investigation. Finally, any external references providing additional, supplemental information about the data should also be included.

The labeling of *context products* for a bundle is the mechanism by which this contextual information is identified for use in the interpretation of science data.

The PDS Engineering Node curates context products for the entire PDS. Thus, when a new mission or investigation is developed, team members from that mission or investigation are responsible for identifying and labeling relevant context products. The product labels for the context products are then submitted to the Engineering Node, from which they can be acquired by other team members and/or investigations.

Much context information evolves during the course of a mission and necessitates updates to the context documentation. For example, unexpected events which occur during the course of a mission will necessitate updates documenting the impact of such changes to instrument and spacecraft systems or to the quality of the data. This might include information such as 'safing' incidents, instrument anomalies or failures, extensions of the mission, etc. All of these updates should be submitted to the Engineering Node.

A context product, like all other PDS products, consists of a data object and an information object. The data object may be a physical object, such as a planet, spacecraft, instrument, or person, or it may be a conceptual object such as a space agency, a mission, or an investigation. The information object consists of a PDS XML label.

A context product is described using the Product_Context class. A context product label is structured much like any other PDS product, beginning with an XML prolog and root element (Fig. 7.1). This preamble is followed by other standard label areas: an Identification_Area, one or more optional Discipline_Areas, an optional Reference_List, and a data object description class.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model
    href="http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.sch"
    schematypens="http://purl.oclc.org/dsdl/schematron"?>

<Product_Context
    xmlns="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:pds="http://pds.nasa.gov/pds4/pds/v03"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="
        http://pds.nasa.gov/pds4/pds/v03
        http://pds.nasa.gov/pds4/pds/v03/PDS4_PDS_0300a.xsd">

    <Identification_Area>
        <logical_identifier>urn:nasa:pds:context:instrument_host:mpfl</logical_identifier>
        <version_id>1.0</version_id>
        <title>MARS PATHFINDER LANDER</title>
        <information_model_version>0.3.0.0.a</information_model_version>
        <product_class>Product_Context</product_class>
        <Modification_History>
            <Modification_Detail>
                <modification_date>2012-10-03</modification_date>
                <version_id>1.0</version_id>
                <description>Adapted from PDS3 insthost.cat for Mars Pathfinder
                    mission.</description>
            </Modification_Detail>
        </Modification_History>
    </Identification_Area>

    ...

</Product_Context>
```

Figure 7.1: A sample preamble and Identification_Area of a context product.

Like all other PDS products, a context product must contain an Identification_Area. Instructions on how to construct a logical_identifier for a context product are provided in section 6.2.5. Otherwise, the Identification_Area is populated in much the same way it is for all other PDS products. (See

```xml
<Reference_List>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:context:investigation:mpf</lid_reference>
    <reference_type>instrument_host_to_investigation</reference_type>
    <comment>Reference to Mars Pathfinder mission.</comment>
  </Internal_Reference>
  <Internal_Reference>
    <lid_reference>urn:nasa:pds:context:target:mars_planet</lid_reference>
    <reference_type>instrument_host_to_target</reference_type>
    <comment>Reference to description of planet Mars.</comment>
  </Internal_Reference>
  <External_Reference>
    <reference_text>Cook, R., P. Katemeyn, and C. Salvo, Mars
      Pathfinder Project Mission Plan, JPL Document
      11355, PF-100-MP-02, 90 pp., 1995.</reference_text>
    <description>
      This Mission Plan provides a high-level summary of the events that occur
      during the Mars Pathfinder mission. The mission plan provides the basis for
      development of detailed mission event sequences and is the baseline for the
      performance of the mission. This document does not levy new mission and system
      requirements, but is intended to show how previously defined requirements and
      constraints are combined to define the mission concept.

      The Mission Plan has a secondary function in providing a descriptive narrative
      of the mission and its various components.  Major design features and options
      are documented, particularly as they affect operations.
    </description>
  </External_Reference>
</Reference_List>
```

Figure 7.2: A sample Reference_List for a context product.

section 3.2 for details.)

Context products may contain one or more optional Discipline_Areas. This permits PDS discipline nodes and data providers to include information relevant to context products that falls outside that typically requested by the PDS. Seek guidance from your discipline node for when and how to use this class in context products.

The Reference_List area (Fig. 7.2) is particularly important for context products. This is the primary mechanism by which the relationships among the various context objects are established. Each Internal_Reference in the Reference_List should link to a PDS product, whether another context product, a document product, or some other product. External_References are used to reference documents available external to the PDS. Each section below will indicate which types of references are expected for each type of context product.

The data object class provides the critical identifying information about a context object. The information required in the data object class varies depending upon the type of context product.

More details about each of the context data object classes are provided in the sections below.

For each section, a sample of the context data object description is shown, along with a brief summary of the attributes expected.

Descriptive information about each context object may either be included in the description field for the data object class in the XML label, or it may be contained in a published paper that is referenced from the label. (A document archived with the PDS may be considered to be "published" by the PDS and is permissible for this purpose.) However, the information must be available in at least one of these forms. Details are provided below on the type of descriptive information expected for each type of context product, where relevant.

7.1 Agency

An *agency* is an entity that provides regional or national level governance over nodes within the federated Planetary Data System. NASA and the European Space Agency (ESA) are examples of agencies.

A description of an agency should include the following:

- a brief summary of the role of the agency
- identification of the organization within the agency responsible for archiving planetary data
- contact information for the archiving organization

```
<Agency>
  <name>National_Aeronautics_and_Space_Administration</name>
  <description>The National Aeronautics and Space Administration (NASA) is
    the agency of the United States government that is responsible for
    the nation's civilian space program and for aeronautics and aerospace
    research.

    Within NASA, the Planetary Data System (PDS) is a distributed data
    system that archives data collected by Solar System robotic missions
    and ground-based support data associated with those missions. PDS is
    managed by NASA Headquarters' Planetary Sciences Division.</description>
</Agency>
```

7.2 Observing System

An observing system comprises one or more subsystems used to collect data. Various subsystems include instruments and the structures, either spacecraft- or ground-based that they are mounted on, plus any artificial illumination sources necessary for the data collection. In cases where a human observer is a key component of the observing system (such as an astronomer looking through a telescope), the human observer is also considered a component.

7.2.1 Facility

Facilities include fixed institutions which may serve as hosts to various instruments. For example, a laboratory in which a camera was mounted for pre-flight calibration purposes is considered to be a facility. Similarly, astronomical observatories are considered to be facilities.

TBD

7.2.2 Instrument

An instrument is a physical object that collects data. This can include a wide variety of apparatuses, including (but not limited to) cameras, spectrometers, magnetometers, barometers, particle detectors, radar instruments, etc.

A description of an instrument should include the following:

- a high level overview of the instrument's characteristics
- identification of the scientific objectives as they relate to instrument design
- calibration methods, procedures, and related commentary
- operational considerations, particularly including any special circumstances or events that affect instrument performance
- a general description of the instrument detectors and their characteristics
- a description of the instrument electronics and any internal data processing
- a description of the instrument filters and any associated calibration
- a description of the instrument optics, if applicable
- an overview of operating modes and configurations used in data acquisition
- a description of any logical subsystems of the instrument
- a description of the actual parameters measured by the instrument

```
<Instrument>
  <name>Imager for Mars Pathfinder</name>
  <version_id>flight</version_id>
  <type>Imager</type>
  <description>The IMP is a stereo imaging system...</description>
  <naif_instrument_id>-53001</naif_instrument_id>
  <serial_number>3942929038</serial_number>
</Instrument>
```

name Provides the name of the instrument. (Cf. "title" in the Identification Area, which provides a title for this instrument context product. The title and name attributes may have the same value.) (optional)

version_id This is a version identifier for a specific piece of hardware; it may be used to distinguish among prototype, engineering, and flight models of an instrument, if applicable. Alternatively, a serial number may be used for this purpose (see below). (optional)

type Indicates the kind of instrument. (See Table **??** for a complete list of possible values.)

description Either a detailed description of an instrument and its subsystems or a brief summary or abstract of information available in an alternate document.

naif_instrument_id The identifier for the instrument used by the NAIF node. (optional)

serial_number An identifier for a specific piece of hardware. (See version_id, above.) (optional)

The list of possible reference_types for internal references from an instrument are:

- context_to_associate
- context_to_bundle
- context_to_collection
- context_to_document
- context_to_resource
- instrument_host_to_document
- instrument_host_to_investigation
- instrument_host_to_target

7.2.3 Instrument Host

An instrument host is a generic term covering any kind of platform on which an instrument may be mounted. The PDS currently recognizes three types of platforms: Earth based, Rover, and Spacecraft. An airplane carrying a telescope would be considered an Earth based instrument host. The term "spacecraft" can be used to refer to an orbiter, a "fly-by" spacecraft, or a lander.

A description of an instrument host should include overviews of the following subsystems, where relevant:

- antennas
- attitude and articulation control
- electronics
- entry, descent, and landing
- command, telemetry, and data handling
- instrument platforms
- launch vehicle
- mechanical devices
- mechanical structure
- navigation
- power generation and distribution
- propulsion
- pyrotechnics
- surface mobility
- telecommunications
- temperature control

TBD

7.2.4 Telescope

The term telescope is used to refer to terrestrial, ground-based telescopes.

TBD

7.3 Investigation

In PDS, the term "investigation" is used as a broad catch-all to describe the conceptual environment within which data acquisition occurs. There are several types of investigations, including missions, observing campaigns, individual investigations, and "other" investigations. The distinctions between these different types of investigations are somewhat indistinct; choose whichever type best describes the situation in which your data was acquired.

A mission typically refers to a flight project funded by one or more nations' space programs, involving the contributions of multiple scientists and institutions. It usually utilizes at least one spacecraft with multiple instruments on board.

An observing campaign generally refers to an effort coordinated across multiple observatories, institutions, and/or spacecraft to observe a particular event. Examples of observing campaigns are the Shoemaker-Levy 9 impact with Jupiter and the Saturn ring plane crossings, which have been observed by multiple spacecraft and observatories.

An individual investigation is used to describe a study or series of experiments carried out by one or a small number of investigators. It is typically relatively small in scale.

An "other" investigation might be used to describe a literature search.

TBD

7.4 Node

This information describes the PDS node responsible for curating the archived data. The PDS node is responsible for providing the information.

TBD

7.5 Personnel

7.5.1 PDS Affiliate

A *PDS affiliate* is a person who has an association with the planetary science community and has access to PDS resources not normally allowed to the general public.

TBD

7.5.2 PDS Guest

A *PDS guest* is a person who has an association with the planetary science community and who has the most limited access to PDS resources.

TBD

7.6 Resource

A *resource* is a web resource or interface to PDS data.

TBD

7.7 Target

PDS now recognizes multiple targets for instrument observations including: calibration objects, dust, features, regions, rings, solar system bodies, solar wind, and stars. This information will be provided by the PDS, gleaned from external sources.

TBD

7.8 Other

The *other* class should be used to describe activities involved in the collection of data which are not otherwise modeled.

TBD

Chapter 8

Other Documentation

Supplementary or ancillary reference materials are usually included with archive products to improve their short- and long-term utility. These documents augment the internal documentation of the product labels and provide further assistance in understanding the data products and accompanying materials. Typical archive documents include:

- Flight project documents
- Instrument papers
- Science articles
- Software Interface Specifications (SISs)
- Software user manuals

The PDS criteria for inclusion of a document in the archive are:

1. Would this information be helpful to a data user?

2. Is the material necessary?

3. Is the documentation complete?

In general, the PDS seeks to err on the side of completeness.

Each document to be archived must be saved in a PDS-compliant format[1] and archived as a PDS product. The Product_Document class is used to describe document products. Document products are delivered as part of a document collection.

[1] Where are these documented? – EDR

A flat, human-readable ASCII text version of each document must be included on the volume, although additional versions may be included in other supported formats at the option of the data producer. Flat ASCII text means the file may contain only the standard, 7-bit printable ASCII character set, plus the blank character and the carriage-return and linefeed characters as record delimiters. A file is human-readable if it is not encoded and if any special markup tags which may be included do not significantly interfere with an average users ability to read the file. So, for example, simple HTML files and TeX/LaTeX files with relatively little markup embedded in the text are generally considered human-readable and may, therefore, be used to satisfy the above ASCII text version requirement. Needs to be updated! – EDR

Note that the PDS takes the requirement for complete documentation very seriously. Documents that are essential to the understanding of an archive are considered as important as the data files themselves. Furthermore, including a document in a PDS archive constitutes publication (or re-publication) of that document. Consequently, documents prepared for inclusion in an archive are expected to meet not only the PDS label and format requirements, but also the structural, grammatical and lexical requirements of a refereed journal submission. Documents submitted for archiving which contain spelling errors, poor grammar or illogical organization will be rejected and may ultimately lead to the rejection of the submitted data for lack of adequate documentation.

Include discussion of labeling of document product here.

Chapter 9

Time Standards

TBD from ISO 8601

The following is a place holder and needs to be updated.

Potentially need to add info about handling negative (BC) dates.

PDS has adopted a subset of the International Standards Organization Standard (ISO/DIS) 8601 standard entitled Data Element and Interchange Formats - Representations of Dates and Times, and applies the standard across all disciplines in order to give the system generality.

It is important to note that the ISO/DIS 8601 standard covers only ASCII representations of dates and times.

Note that the "SPICE" system, generally used by planetary missions for computing observation geometry, uses additional time systems and time representations.

9.1 Date/Times

In the PDS there are two recognized date/time formats: Now incorrect!

CCYY-MM-DDTHH:MM:SS.sss (preferred format)
CCYY-DDDTHH:MM:SS.sss

Each format represents a concatenation of the conventional date and time expressions with the two parts separated by the letter T:

CC - century (00-99)
YY - year (00-99)
MM - month (01-12)
DD - day of month (01-31)
DDD - day of year (001-366)
T - date/time separator
HH - hour (00-23)
MM - minute (00-59)
SS - second (00-59)
sss - fractions of second (000-999)

Note: See Section 7.4 Midnight and Leap Seconds for special cases involving the indication of midnight and leap seconds.

The preferred date/time format is: CCYY-MM-DDTHH:MM:SS.sss.

Date/Time Precision

The above date/time formats may be truncated on the right to match the precision of the date/time value in any of the following forms:

1998
1998-12
1998-12-01
1998-12-01T23
1998-12-01T23:59
1998-12-01T23:59:58
1998-12-01T23:59:58.1
1998-12-01T23:59:58.12

9.2 Dates

Dates should be expressed in the conventional ISO/DIS 8601 format. On those rare occasions when dates cannot be expressed in the conventional format, a native format may be used.

Need to discuss negative dates.

9.2.1 Conventional Dates

Conventional dates are represented in ISO/DIS 8601 format as either year (including century), month, day-of-month (CCYY-MM-DD), or as year, day-of-year (CCYY-DDD). The hyphen character (-) is used as the field separator in this format. The year, month, day-of month format is the preferred format for use in PDS labels and catalog files and is referred to as PDS standard date format, but either format is acceptable.

9.2.2 Native Dates

Dates in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the specific data set, thus native dates. Native date formats are specified by the data preparer in conjunction with the PDS data engineer. Mission-elapsed days and time-to-encounter are both examples of native dates.

9.3 Times

The PDS allows times to be expressed in conventional and native (alternate) formats.

9.3.1 Conventional Times

Conventional times are represented as hours, minutes and seconds according to the ISO/DIS 8601 time format standard: HH:MM:SS[.sss]. Note that the hours, minutes, and integral seconds fields must contain two digits. The colon (':') is used as a field separator. Fractional seconds consisting of a decimal point (the European-style comma may not be used) and up to three digits (thousandths of a second) may be included if appropriate.

Coordinated Universal Time (UTC) is the PDS time standard and must be formatted in the previously described ISO/DIS 8601 standard format. The letter "Z", indicating the civil time zone at Greenwich (i.e., GMT), should never be appended to a UTC time. The relationship between UTC and GMT has varied historically and with observer context. Note that in PDS data sets created under earlier versions of the Standards, an appended Z is taken as indicating UTC.

The start_time and stop_time attributes required in data product labels are in UTC. For data collected by spacecraft-mounted instruments, the date/ time must be a time that corresponds to spacecraft event time. For data collected by instruments not located on a spacecraft, this time shall be an earth-based event time.

Adoption of UTC (rather than spacecraft-clock-count, for example) as the standard facilitates comparison of data from a particular spacecraft or ground-based facility with data from other sources.

9.3.2 Native Times

Times in any format other than the ISO/DIS 8601 format described above are considered to be in a format native to the data set, and thus native times. Data preparers should consult a PDS data engineer for assistance in selecting appropriate PDS attributes for expressing native times.

Examples of quantities that may be expressed in native time formats are included below.

9.3.2.1 Spacecraft Clock Count (sclk)

There is one native time of particular interest, however, which has specific keywords associated with it. The spacecraft clock reading (that is, the count) often provides the essential timing information for a space-based observation. Therefore, the attributes spacecraft_clock_start_count and spacecraft_clock_stop_count are required in labels describing space-based data. This value is formatted as a string to preserve precision.

9.3.2.2 Ephemeris Time

Need to talk to Chuck and/or Boris to determine if they want to say anything here.

9.3.2.3 Relative Time

9.3.2.4 Local Solar Time

9.3.2.4.1 Local True Solar Time

The local_true_solar_time attribute describes the local true solar time (LTST). It is one of two types of solar time used to express the time of day at a point on the surface of a planetary body. LTST is measured relative to the true position of the Sun as seen from a point on the planet's surface. The coordinate system used to define LTST has its origin at the center of the planet. Its Z-axis is the north pole vector (or spin axis) of the planet. The X-axis is chosen to point in the direction of the vernal equinox of the planet's orbit. (The vernal or autumnal equinox vectors are found by searching the planetary ephemeris for those times when the vector from the planet's center to the Sun is perpendicular to the planet's north pole vector. The vernal equinox is the time when the Sun appears to rise above the planet's equator.) Positions of points in this frame can be expressed as a radius and planetocentric 'right ascension' and 'declination' angles. The planetocentric right ascension angle, or RA, is measured positive eastward in the equatorial plane from the vernal equinox vector to the intersection of the meridian containing the point with the equator. Similarly, the planetocentric declination is the angle between the equatorial plane and the vector to the point. LTST is a function of the difference between the RAs of the vectors to the Sun and to the point on the planet's surface. Specifically,

LTST = (a(P) - a(TS)) * (24 / 360) + 12

where,

LTST = the local true solar time in true solar hours
a(P) = RA of the point on the planet's surface in deg
a(TS) = RA of the true sun in deg

The conversion factor of 24/360 is applied to transform the angular measure in decimal degrees into hours-minutes-seconds of arc. This standard representation divides 360 degrees into 24 hours, each hour into 60 minutes, and each minute into 60 seconds of arc. The hours, minutes, and seconds of arc are called 'true solar' hours, minutes, and seconds when used to measure LTST. The constant offset of 12 hours is added to the difference in RAs to place local noon (12:00:00 in hours, minutes, seconds) at the point where the Sun is directly overhead; at this time, the RA of the true sun is the same as that of the surface point so that a(P) - a(TS) = 0. The use of 'true solar' time units can be extended to define a true solar day as 24 true solar hours. Due to the eccentricity of planetary orbits and the inclination of orbital planes to equatorial planes (obliquity), the Sun does not move at a uniform rate over the course of a planetary year. Consequently, the number of SI seconds in a true solar day, hour, minute or second is not constant. (Definition adapted from Vaughan (1995).)

9.3.2.4.2 Local Mean Solar Time

The desire to work with solar days, hours, minutes, and seconds of uniform length led to the concept of the fictitious mean Sun (FMS). The FMS is defined as a point that moves on the celestial equator of a planetary body at a constant rate that represents the average mean motion of the Sun over a planetary year. local mean solar time (LMST), is defined, by analogy with LTST, as the difference between the planetocentric right ascensions of a point on the surface and of the FMS.

The difference between LTST and LMST varies over time. The length of a mean solar day is constant and can be computed from the mean motion of the FMS and the rotation rate of a planet. The mean solar day is also called a 'sol'. Mean solar hours, minutes, and seconds are defined in the same way as the true solar units.

The acceptable range of values for local_mean_solar_time is '00:00:00.000' to '23:59:59.999'. (Definition adapted from Vaughan (1995).)

9.4 Midnight and Leap Seconds

The ISO/DIS 8601 standard for representation of midnight and leap seconds are also used in PDS time fields.

9.4.1 Midnight

Midnight may be indicated in one of two ways: as 00:00:00 or 24:00:00. The usual precision modifications apply as well i.e. 24:00 is also recognized as midnight.

The 00:00:00 notation is used to indicate midnight at the beginning of a date. 24:00:00 is used to indicate midnight at the end of a date. So, for example, the following two date/time strings refer to precisely the same moment:

2007-04-07T24:00:00 = 2007-04-08T00:00:00

When the hours field has the value 24, any and all subsequent time fields must be zero.

9.4.2 Leap Seconds

Leap seconds may be positive or negative, but in either case are always applied at the end of the day in question. A positive leap second is indicated with a time value of 23:59:60. A negative leap second is indicated by the omission of the time 23:59:59. That is, on the day of a negative leap second, the sequence leading through midnight is:

23:59:57
23:59:58
00:00:00
00:00:01

And on the day of a positive leap second, the sequence through midnight is:

23:59:58
23:59:59
23:59:60
00:00:00
00:00:01

Note that the only time when the seconds value of a time string may contain the value 60 is when this represents a positive leap second.

Chapter 10

Units of Measurement

The uniform use of units of measure facilitates broad catalog searches across archive systems.The PDS standard system for units, where applicable, is the International System of Units (SI). The default units for data elements in the Planetary Science Data Dictionary (PSDD) are determined as each element is defined and added to the dictionary. Specific unit definitions are also included in the PSDD.

The PDS allows exceptions to the SI unit requirement when common usage conflicts with the SI standard (e.g., angles which are measured in degrees rather than radians).

Both singular and plural unit names, as well as unit symbols, are allowed. The double asterisk (**) is used, rather than the caret (ˆ), to indicate exponentiation. When the units associated with a value of a PDS element are not the same as the default units specified in the PSDD (or when explicit units are preferred), a unit expression is used with the value. These unit expressions are enclosed in angular brackets (¡ ¿) and follow the value to which they apply.

Update to indicate implementation using attributes.

Examples Needs to be udpated

EXPOSURE_DURATION = 10 seconds
DECLINATION = -14.2756 degrees
MASS = 123 kg
MASS_DENSITY = 123 g/cm^3
MAP_RESOLUTION = 123 pixel/degree
MAP_SCALE = 123 km/pixel

Note that in the above example, MASS_DENSITY is not expressed in the SI default unit of measurement for density (kg/m^3).

PDS recommends (in order of preference) that measurements be expressed using the default SI units of measurements, as defined in the following paragraphs. If it is not desirable to use the default SI unit of measurement, then the unit of measurement should be expressed using the SI nomenclature defined in the following paragraphs. If a unit of measurement is not defined by the SI standard, then a unit of measurement can be derived (e.g., pixels per degree, kilometers per pixel, etc.).

10.1 SI Units

The following summary of SI unit information is extracted from The International System of Units.

Base units As the system is currently used, there are seven fundamental SI units, termed base units:

QUANTITY	NAME OF UNIT	SYMBOL
length	meter	m
mass	kilogram	kg
time	second	s
electric current	ampere	A
thermodynamic temperature	kelvin	K
amount of substance	mole	mol
luminous intensity	candela	cd

SI units are all written in mixed case; symbols are also mixed case except for those derived from proper names. No periods are used in any of the symbols in the international system.

Derived units In addition to the base units of the system, a host of derived units, which stem from the base units, are also employed. One class of these is formed by adding a prefix, representing a power of ten, to the base unit. For example, a kilometer is equal to 1,000 meters, and a millisecond is .001 (that is, 1/1,000) second. The prefixes in current use are as follows:

SI PREFIXES

Factor	Prefix	Symbol	Factor	Prefix	Symbol
10^{18}	exa	E	10^{-1}	deci	d
10^{15}	peta	P	10^{-2}	centi	c
10^{12}	tera	T	10^{-3}	milli	m
10^{9}	giga	G	10^{-6}	micro	
10^{6}	mega	M	10^{-9}	nano	n

10^3	kilo	k	10^{-12}	pico	p
10^2	hecto	h	10^{-15}	femto	f
10^1	deka	da	10^{-18}	atto	a

Note that the kilogram (rather than the gram) was selected as the base unit for mass for historical reasons. Notwithstanding, the gram is the basis for creating mass units by addition of prefixes.

Another class of derived units consists of powers of base units and of base units in algebraic relationships. Some of the more familiar of these are the following:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
area	square meter	m^2
volume	cubic meter	m^3
density	kilogram per cubic meter	kg/m^3
velocity	meter per second	m/s
angular velocity	radian per second	rad/s
acceleration	meter per second squared	m/s^2
angular acceleration	radian per second squared	rad/s^2
kinematic viscosity	square meter per second	m^2/s
dynamic viscosity	newton-second per square meter	$N * s/m^2$
luminance	candela per square meter	cd/m^2
wave number	1 per meter	m^{-1}
activity (of a radioactive source)	1 per second	s^{-1}

Many derived SI units have names of their own:

QUANTITY	**NAME OF UNIT**	**SYMBOL**	**EQUIVALENT**
frequency	hertz	Hz	s^-1
force	newton	N	$kg * m/s^2$
pressure (mechanical stress)	pascal	Pa	N/m^2
work, energy, quantity of heat	joule	J	$N * m$
power	watt	W	J/s
quantity of electricity potential difference	coulomb	C	$A * s$
electromotive force	volt	V	W/A
electrical resistance	ohm		V/A
capacitance	farad	F	$A * s/V$
magnetic flux	weber	Wb	$V * s$
inductance	henry	H	$V * s/A$
magnetic flux density	tesla	T	Wb/m^2

luminous flux	lumen	lm	$cd * sr$
illuminance	lux	lx	lm/m^2

Supplementary units are as follows:

QUANTITY	**NAME OF UNIT**	**SYMBOL**
plane angle	radian	rad
solid angle	steradian	sr

Use of figures with SI units In the international system it is considered preferable to use only numbers between 0.1 and 1,000 in expressing the quantity associated with any SI unit. Thus the quantity 12,000 meters is expressed as 12 km, not 12,000 m. So too, 0.003 cubic centimeters is preferably written 3 mm3, not 0.003 cm3.

Part III

Discipline Standards

Chapter 11

Calibration Standards

TBD - This section is targeted for completion in the second half of calendar year 2012.

Chapter 12

Cartographic Standards

Note - Geometric and cartographic standards are still under development and are targeted for completion in the second half of calendar year 2012.

12.1 Introduction

To facilitate use, exchange and integration of its products, the PDS follows accepted planetary cartographic standards for data products where they exist. Because such standards evolve as new data and knowledge are acquired, there are advisory groups charged with developing and periodically updating standards for coordinate systems. All data providers for PDS products should follow accepted standards and be aware of current NASA and international recommendations on cartographic coordinate systems and conventions relevant to their bodies of interest. An absolute requirement for all PDS products is that relevant coordinate systems and frames be clearly specified in product labels and supporting documents. This chapter specifies, as of early 2011, the authoritative sources for international cartographic standards, provides a summary of major cartographic elements to which those standards apply, and identifies the primary standards that PDS has adopted.

12.1.1 International and NASA Advisory Groups for Cartographic Standards

The primary international body for coordinate systems in the Solar System is the International Astronomical Union (IAU). The IAU has recognized the International Celestial Reference System

(ICRS) as the defining inertial reference system and its associated International Celestial Reference Frame (ICRF) (Ma et al., 1998) as the defining frame for that system. The ICRS and ICRF are maintained for the IAU by the International Earth Rotation and Reference Systems Service (IERS, http://www.iers.org/).

For cartographic coordinates and conventions for planets and satellites, the IAU and the International Association of Geodesy (IAG) have established jointly the Working Group on Cartographic Coordinates and Rotational Elements (WGCCRE), which publishes triennial reports, currently in the journal *Celestial Mechanics and Dynamical Astronomy* (Davies et al., 1980, 1983, 1986, 1989, 1992, 1996; Seidelmann et al., 2002, 2005, 2007; Archinal et al., 2011). This working group includes PDS-affiliated scientists, thus assuring full interaction in defining the standards. Publications and reports issued by the WGCCRE can be found at http://astrogeology.usgs.gov/Projects/WGCCRE/. PDS data providers should refer to these reports for current information and recommendations on rotational elements for Solar System bodies and how these are related to their cartographic coordinates.

The NASA Lunar Geodesy and Cartography Working Group (LGCWG) and the Mars Geodesy and Cartography Working Group (MGCWG) are sponsored by the NASA Lunar Precursor Robotics Program (LPRP) and Mars Program offices, respectively, and are responsible within NASA for providing additional coordination of cartographic standards and related (e.g., data processing) issues (Archinal et al., 2008a; Archinal et al., 2008b; Duxbury et al. 2002). These Working Groups have made additional recommendations regarding coordinate systems (generally with additional detail) beyond those of the WGCCRE.

12.2 Inertial Reference Frame and Time System

The orientation of a body in the Solar System can be calculated using a series of rotation angles to define the directions of the body's principal axes with respect to an inertial reference frame (i.e., a system that is not rotating or accelerating relative to a specific reference point) which provides a standard frame from which position, velocity, and acceleration can be measured. Such a reference frame is a set of identifiable fiducial points and their positions on the sky, providing a practical realization of a reference system that defines the origin, fundamental planes (or axes), and transformations between observed elements and reference points in the celestial coordinate system. Reference coordinate systems are defined by a system of concepts (e.g., using planetocentric latitude and longitude) while a reference coordinate frame is a specific realization of a coordinate system that is anchored to real data (such as a photogrammetric control network, altimetry crossover solutions, or lunar ephemerides) (Kovalevsky and Mueller, 1981).

For a planetary body in space, position is defined relative to a Z axis (typically the spin vector of the body, or the planetographic north pole), the X axis (defined as the point where the equator of the body crosses the equatorial plane of an inertial frame at a specific epoch), and the Y axis of a right-handed system. The standard units for coordinates are based on the SI, including decimal degrees. The orientation of Solar System bodies can be calculated from angular position (right ascension α and declination δ) with respect to the equatorial system of a particular epoch. For example, the orientation of the north pole of a body at a given epoch is specified by its right ascension α and declination δ, while the location of the prime meridian is specified by the angle W (Davies et al., 1980).

The standard epoch is called J2000.0 and is defined to be 2000 January 1.5 Barycentric Dynamical Time (TDB) (e.g., Seidelmann et al., 2007). This corresponds to 2000 January 1, 1200 hours Terrestrial Time (TT) or the Julian Date 2451545.0 (NAO, USNO and HMNAO, 1983). This also corresponds to 2000 January 1, 11:58:55.816 Coordinated Universal Time (UTC) (Seidelmann et al., 1992). Although the natural system for many applications would be TDB, UTC is considered the fundamental system for all PDS data products. The standard way of expressing UTC is in year, month, day, hour, minute, and decimal seconds. Julian Dates (JD) are supported as a supplementary system for reporting UTC time. However the JD time scale must be specified (e.g., UTC or TDB). See chapter 9 of this document for further information on time representation.

The currently accepted orientation of the inertial system (i.e., J2000.0 right ascension and declination) is defined by the International Celestial Reference System (ICRS), which is a particular implementation of the Barycentric Celestial Reference System (BCRS) (IAU, 2002). The ICRS is the fundamental celestial reference system of the IAU, and it has an origin at the barycenter of the Solar System and space fixed (kinematically non-rotating) axis directions. As noted by the IAU, the ICRS is meant to represent the most appropriate coordinate system for expressing reference data on the positions and motions of celestial objects. Specifications for the ICRS include a metric tensor, a prescribed method for establishing and maintaining axis directions, a list of benchmark objects with precise coordinates, and standard algorithms to transform these coordinates into observable quantities for any location and time. The ICRS is derived from the International Celestial Reference Frame (ICRF) comprised of coordinates for a set of fiducial points on the sky. The ICRF is within 0.05 arcseconds (Chapront et al., 2002; Herring et al., 2002) of the Solar System inertial frame based on Earth's Mean Equator (EME) at the Equinox of Julian Ephemeris Date (JD) 2451545.0 (i.e., J2000.0). This is consistent with current dynamical practice and spacecraft and planetary ephemerides (e.g., those provided by the NASA Jet Propulsion Laboratory).

Many older data sets, collected before the J2000.0 system and ICRF were defined, are referenced to EME and Equinox of Besselian 1950.0 (B1950.0; JD 2433282.423). While this reference frame should not be used for current data, PDS supports this reference frame for older data. Transformation between the B1950.0 and J2000.0 (and the nearly equivalent ICRF) systems has been well defined by the IAU (NAO, USNO and HMNAO, 1983; also see http://nedwww.ipac.caltech.edu/forms/calculator.html).

Positions may be expressed in other coordinate systems and associated frames, which can be derived from the fundamental system and frame, when this enhances the use of the data for various applications. These include ecliptic-based coordinates and heliographic coordinates. These coordinates, while possibly "natural" for many applications, are derivable from the fundamental system and are therefore treated as supplementary data by PDS. In some cases, it is convenient to work in one preferred coordinate system and then to convert to another, more standard system for products. This practice of providing the natural working coordinates in addition to the coordinates in a fundamental system promotes ease of use of PDS products and should be adopted by all data providers who use coordinate systems other than the fundamental system. As noted above, all supplementary coordinate systems must be fully documented in PDS products and must be negotiated with the PDS prior to delivery.

12.3 Spin Axes and Prime Meridians

The spin axis orientations of many Solar System bodies are defined by the WGCCRE in the ICRF inertial reference frame. For historical reasons, the orientation of the spin axis of planets and satellites is defined by the north pole, which is the pole that is on the northern side of the Invariant Plane of the Solar System (close to but not the same as the ecliptic). With this definition of the north pole, it is also necessary to specify whether the rotation is direct or prograde (in the same direction as the Sun's rotation or counterclockwise when viewed from above the north pole) or retrograde (opposite to the direction of the Sun's rotation).

For small bodies such as comets and asteroids, for which precession due to torques can cause large changes in the angular momentum vector, the orientation is defined by the positive pole, which is the pole determined by the right hand rule for rotation. Since some small bodies can be in excited state rotation, there are numerous complications in application that are addressed in more detail in the WGCCRE reports. Depending on the mode of excited state rotation, the axis may coincide with the maximum moment of inertia. Some cases, particularly the case of chaotic rotation, are considered on a case by case basis by the WGCCRE.

If a body has a solid surface, prime meridians for a given longitude system may be defined by specifying the coordinates of a surface feature on the body (usually a small feature such as a crater in the equatorial region) or by the mean direction relative to the parent body for synchronously rotating bodies (e.g., the Moon, the Galilean moons, and most of the Saturnian moons). Where insufficient observations exist to determine the principal moment of inertia, coordinates of a surface feature will be specified and used to define the prime meridian. In the case of planets without solid surfaces, the definition of the prime meridian is somewhat arbitrary. In any case, the actual definitions are decided by the WGCCRE, not by the PDS. We note that influxes of new data often lead to an iterative process to define (or improve) the orientation of the spin axis or other parameters used to define a coordinate system and in these cases the data providers (e.g., spacecraft mission personnel) and the WGCCRE must maintain close contact regarding the definition.

12.4 Body-Fixed Planetary Coordinate Systems

Two types of coordinate systems are fixed to the body planetocentric and planetographic. Details of the coordinate systems for planets and satellites differ from those for small bodies and rings. This section discusses only the aspects that are common to all applications. The Planetocentric system has an origin at the center of mass of the body. Planetocentric coordinates are defined by a vector from the center of mass of the body (often approximated as the center of figure) to the point of interest, typically but not necessarily a point on the surface (e.g., an impact crater with known position). The planetocentric latitude is the angle between the equatorial plane and the vector, while the planetocentric longitude is the angle between the prime meridian and the projection of the vector onto the equatorial plane.

The Planetographic system also has an origin at the center of mass of the body. Planetographic coordinates, however, are defined by vectors perpendicular to a reference surface, often a biaxial ellipsoid that is centered on the body and chosen to describe the gross shape of the body. Reference surfaces vary from body to body and are defined by the WGCCRE in consultation with the observers who provide the information to define such surfaces. The most common reference surface is an oblate spheroid aligned with the spin axis of the body. However, for certain applications the reference surface may be a triaxial ellipsoid, a gravitational equipotential, or a higher order surface model.

For a biaxial ellipsoid the planetographic latitude is the angle between the equatorial plane and a vector through the point of interest, where the vector is normal to the reference surface. Planetographic longitude is the angle between the prime meridian and the projection of the same vector onto the equatorial plane. In general, the planetographic vector does not pass through the origin. The vector need not pass through the spin axis but in most realistic cases it does. If the reference surface is a sphere, the planetographic and planetocentric vectors are identical.

The WGCCRE allows for the use of either planetographic or planetocentric coordinates for a given body, so data providers may adopt either system. Historically planetographic coordinates have been preferred for cartographic products, while planetocentric coordinates were used for dynamical (i.e. orbit, gravity field, altimetric) observations and calculations. For the planet Mercury, the MESSENGER mission has chosen to use planetocentric coordinates as the primary coordinate system for all products (Seidelmann et al., 2007). For the planet Mars, the MGCWG and all current NASA missions have chosen to use planetocentric coordinates as the primary coordinate system for products (Duxbury et al., 2002). Producers of printed or electronically printed maps (e.g., in PDF format) may wish to show both types of coordinates.

12.4.1 Planets and Satellites

For planets and satellites, the conventions are complicated for historical reasons. In the planetocentric coordinate system, northern latitudes are those in the hemisphere of the body containing the spin pole that points to the northern side of the invariant plane of the Solar System. The body's rotation direction, either prograde or retrograde, must also be specified. Planetocentric longitude increases eastward (i.e., in the direction defined by the right-hand rule and the north pole) from the prime meridian, from 0° to 360°. Thus an external observer sees the longitude decreasing with time if the rotation is prograde but increasing with time if the rotation is retrograde.

North and south planetographic latitude are defined in the same way as for planetocentric latitude, although the numerical values for a given point on the surface, (other than on the equator or at the poles) are different if the reference surface is not a sphere. The definition of planetographic longitude is dependent upon the rotation direction of the body, with the basic definition being that an external observer should see the longitude increasing with time, or that the longitude increases in the direction opposite to the rotation, although there are exceptions due to historical practice for Earth, the Moon, and the Sun. That is to say, the longitude increases to the west if the rotation is prograde (or eastward) and vice versa. Whether the rotation direction is prograde or retrograde can be determined from the current WGCCRE report. See Tables 1 and 2 (or their equivalent in any future report), where the sign of the velocity term for W indicates either prograde (positive) or retrograde (negative) rotation. For all bodies a longitude range of 0° to 360° can be used.

For Earth, the Moon, and the Sun, a longitude range of -180° to +180° has been used in the past (including in existing sets of data archived with the PDS, as defined by the Planetary Science Data Dictionary (PDS, 2002) (fix reference - EDR) and is allowed by the WGCCRE. However, for the Moon, the NASA LGCWG and Lunar Reconnaissance Orbiter (LRO) Mission recommend that in the future, only the 0° to 360° range be used (LGCWG, 2008; LRO Project, 2008). For printed or electronically printed maps (e.g., in PDF format), it may be useful to label the longitude grid both with primary 0° to 360° coordinates and -180° to +180° coordinates.

For the Moon, two slightly different reference systems are commonly used to orient the lunar body-fixed coordinate system. One is the Mean Earth/Polar Axis (ME) system, the preferred system to be used for PDS data products. The other is the axis of figure system, also called the Principal Axis (PA) system, sometimes used internally among instrument teams for specific applications. For computing precise lunar coordinates, the WGCCRE recommends the use of the Jet Propulsion Laboratory (JPL) DE403 ephemeris (which provides lunar orientation in the PA system), rotated into the ME system. The WGCCRE noted in its most recent report that improved versions of the JPL ephemerides were imminent and might be used instead. In fact the JPL DE421 ephemeris is now available and, after rotation into the ME system, is recommended for use (LGCWG, 2008; LRO Project, 2008). The maximum difference between these two frames in the ME system for the period 2000-2019 is only about 6 meters (Archinal, 2008).

12.4.2 Small Bodies

For small bodies (asteroids and comets), both planetographic and planetocentric coordinates follow the same right hand rule that is used to define the positive pole, which can be either above or below the invariant plane of the Solar System. For the simple case of a body with positive pole pointing to the northern hemisphere of the Solar System, this corresponds to longitude, both planetocentric and planetographic, increasing eastward, 0° to 360°, which in turn corresponds to the case in which the longitude seen by an outside observer decreases with time.

For some small bodies, coordinates based on latitude and longitude alone can be multi-valued in radius i.e., the vector from the center of the body can intersect the surface in more than one place. There may also be complications (due to the irregular shape) which force special procedures when producing a useful, planar map. Such details are discussed in reports of the WGCCRE.

12.4.3 Rings

There is no international standard for ring coordinate systems. Standards in use for such PDS products were defined by experts in the Rings Node, in consultation with a broad cross-section of interested scientists. Conventions for coordinate systems for rings are similar to those for small bodies, in as much as they are all based on a right-hand rule, with longitude increasing in the direction of orbital motion. Thus longitude increases eastward for the prograde-moving rings (Jupiter, Saturn, and Neptune), but it increases westward for retrograde-moving rings of Uranus. Rings also use a positive pole direction following the right hand rule, analogous to the case for small-body rotation, thus coinciding with the North Pole of Jupiter, Saturn, and Neptune, but the South Pole of Uranus.

Coordinates for rings differ from those for planets and small bodies in not being body-fixed because there are no fixed features to define longitude. They are defined in an inertial system that is co-moving with the center of mass of the parent body. Specifically, longitudes are measured from the ascending node of the plane of the rings in the ICRF, i.e. the point at which the plane of the rings intersects the ICRF equator. In the case of inclined rings, longitudes are measured as a broken angle from the ascending node of the planets equatorial plane in the ICRF, along the equatorial plane to the ring planes ascending node, and thereafter along the ring plane.

12.4.4 Planetary Plasma Interactions

There are no international standards for values or names of coordinate systems of planetary plasma observations. Recommendations for coordinate systems in the near-Earth environment by Russell

(1971) have been generalized for use with plasma observations at other bodies. More recently, other systems have been defined (e.g., Franz and Harper, 2002) and are currently in use. The coordinate systems used for plasma observations and data analysis typically are right-handed. The primary exception to this rule is the left-handed Jovian System III.

Standards for planetary plasma data products for PDS were defined by experts in the Planetary Plasma Interactions (PPI) Node, following recommendations from Russell (1971) and Franz and Harper (2002) and in consultation with other specialists. Providers and users of PDS data featuring plasma observations are encouraged to use names as defined by these authors where appropriate, and to follow similar name construction when new systems must be defined.

12.5 Surface Models

A standard reference surface model commonly used for hard surfaces is the digital terrain model (DTM). The DTM defines body radius or geometric height above the body reference surface as a function of cartographic latitude and longitude. Spheroids, ellipsoids and harmonic expansions giving analytic expressions for radius as a function of cartographic coordinates are all allowed in PDS. A DTM may also define potential height, i.e., elevation, above an equipotential surface, provided the method is specified, including the specification of appropriate constants and gravity field that is used to convert to/from radii and potential height.

The only internationally recognized DTM is the MOLA model for Mars (Seidelmann et al., 2007, page 168 in WGCCRE #10). (WGCCRE #10 is (Archinal et al., 2011), which has only been published online to date, so I'm not sure where the "page 168" comes from. Need to clarify this with Lisa G. - EDR) DTMs are also available for other bodies, including the Moon and several small bodies; but their use is not officially recommended and therefore up to the individual user.

The digital image model (DIM) defines body brightness in a specified spectral band or bands as a function of cartographic latitude and longitude. A DIM may be associated with the surface radius, geometric height, or potential height values in a corresponding DTM or it may be registered independently to a spheroid, ellipsoid, or spherical harmonic expansion.

12.6 PDS Keywords for Cartographic Coordinates

NOTE: Work is currently ongoing on defining the cartographic classes to be used in PDS4; drafts of these are expected to be available roughly mid-year, 2011. When they become available, this section will be updated. - EDR

To support the descriptions of these various reference coordinate systems and frames, the PDS has defined the following set of geometry data elements [see the Planetary Science Data Dictionary (PDS, 2008) (update ref - EDR) for complete definitions and additional data elements].

A_AXIS_RADIUS
B_AXIS_RADIUS
C_AXIS_RADIUS
COORDINATE_SYSTEM_CENTER_NAME
COORDINATE_SYSTEM_DESC
COORDINATE_SYSTEM_ID
COORDINATE_SYSTEM_NAME
COORDINATE_SYSTEM_REF_EPOCH
COORDINATE_SYSTEM_TYPE
EASTERNMOST_LONGITUDE
LATITUDE
LONGITUDE
MAXIMUM_LATITUDE
MAXIMUM_LONGITUDE
MINIMUM_LATITUDE
MINIMUM_LONGITUDE
POSITIVE_LONGITUDE_DIRECTION
WESTERNMOST_LONGITUDE

To support the description of locations in a planetary ring system, the PDS has defined the following data elements:

CENTER_RING_RADIUS
RING_RADIUS
MINIMUM_RING_RADIUS
MAXIMUM_RING_RADIUS

RING_LONGITUDE
MINIMUM_RING_LONGITUDE
MAXIMUM_RING_LONGITUDE

B1950_RING_LONGITUDE
MINIMUM_B1950_RING_LONGITUDE
MAXIMUM_B1950_RING_LONGITUDE

RING_EVENT_TIME
RING_EVENT_START_TIME
RING_EVENT_STOP_TIME

RADIAL_RESOLUTION
MINIMUM_RADIAL_RESOLUTION
MAXIMUM_RADIAL_RESOLUTION

The radius and longitude elements define an inertial location in the rings, and the ring event time elements define the time at the ring plane to which an observation refers. If desired, the radial resolution elements can be used to specify the radial dimensions of ring features that can be resolved in the data. See the Planetary Science Data Dictionary (PSDD; PDS, 2008) for complete definitions of these elements.

Some rings are not circular and/or equatorial. In these cases, the PSDD provides additional elements that can be used to describe a rings shape. The elements are:

RING_SEMIMAJOR_AXIS
RING_ECCENTRICITY
RING_PERICENTER_LONGITUDE
PERICENTER_PRECESSION_RATE
RING_INCLINATION
RING_ASCENDING_NODE_LONGITUDE
NODAL_REGRESSION_RATE
REFERENCE_TIME

Here the value of REFERENCE_TIME indicates the instant at which the LONGITUDE elements are defined. The actual pericenter and ascending node at the time of an observation are determined based on the precession and regression rates as follows:

```
pericenter_longitude = RING_PERICENTER_LONGITUDE +
        PERICENTER_PRECESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
  ascending_node_longitude =
        RING_ASCENDING_NODE_LONGITUDE +
        NODAL_REGRESSION_RATE *
        (observation_time - REFERENCE_TIME) mod 360
```

The oscillating modes of a ring can also be specified if necessary:

RING_RADIAL_MODE
RING_RADIAL_MODE_AMPLITUDE
RING_RADIAL_MODE_FREQUENCY
RING_RADIAL_MODE_PHASE

Additional elements should be used to specify the assumed orientation of the planets pole:

POLE_RIGHT_ASCENSION
POLE_DECLINATION
COORDINATE_SYSTEM_ID

The COORDINATE_SYSTEM_ID can be either J2000.0 or B1950.0, with J2000.0 serving as the default. See the PSDD for further details.

12.7 Map Resolution

A uniform set of resolutions is helpful for analyses of multiple datasets and development of map products derived from PDS data, and the selected scale must account for differences in available image resolution and quality. Such map scales are measured against a reference surface that is typically a geometrically defined shape that represents a given planetary body. For global maps, the recommended spatial resolution for a map is 2^n pixels per degree of latitude, where a pixel is treated as a finite area and n is an integer. A spatial resolution of 2^n pixels per degree allows simple co-registration of multiple datasets by doubling or halving the pixel sizes (typically by averaging or interpolation) and without resampling or otherwise changing the pixels. These recommendations continue a convention established in the 1960s and 1970s by the lunar and Mars research communities (e.g., Batson, 1987; Greeley and Batson, 1990), as advocated by the NASA Planetary Cartography Working Group (PCWG) and its successor the Planetary Cartography and Geologic Mapping Working Group (PCGMWG) (PCWG, 1993, pp. 22-24), and affirmed by the LGCWG (2008).

For polar regions of global maps, the recommendation is also to use the binary map scale or 2^n pixels per degree of latitude near the pole. This practice maintains consistency with the global data product.

For working at landing site scales with data that has pixels of tens of centimeters to a few meters in size, spatial resolutions of maps are more convenient if provided at scales of 1 meter per pixel resolution or multiples thereof (LGCWG, 2008). At such human scales this convention is simpler and will preserve inherent details of resolution for applications such as landing site operations, traversing, and surface engineering studies.

For both global and local maps showing elevation or relief, the recommended vertical resolution is 1 x 10^m meters, where m is an integer chosen to preserve all the resolution inherent in the data.

Need to determine desired format for references in Bibliography and design or acquire a .bst file to accomplish that. - EDR

Need to determine how to output dois, if available. (Already done for some doc. types, but not others?) - EDR

How to include statement at beginning of section 2.8 in StdRefv3.8 about WGCCRE Report 7 not being issued? This needs to be inserted after the ”Bibliography” statement, but before the list of references. - EDR

I’ve dropped the ”chair”, ”vice-chair” and ”consultant” parenthetical statements from the WGCCRE report references; how important is it to have them? (Can be done, but messes up the citations. Probably fixable with research.) - EDR

Need to fix Archinaletal2008a reference; currently missing name of the ”event”. Also need to determine how to include URLs in references. - EDR

Need to fix Archinaletal2008b reference to include abstract number and URL. - EDR

For all WGCCRE reports, need to determine how to append report number. - EDR

Need to fix Duxburyetal2002a reference to include meeting name and URL. - EDR

Need to fix GreeleyBatson1990a reference to include number of book pages. - EDR

Need to include URL for IAU2002a. - EDR

I’ve added information for KovalevskyMueller1981a; is this okay? - EDR

LGCWG2008a - need to add URL; also, how to indicate draft (or find published version)? - EDR

LRO2008a - need to include verion. - EDR

PCWG1993a only seems to be available as one used copy on Amazon for $100; is this still publicly available? - EDR

Need to add PDS4 PSDD reference. - EDR

In Seidelmanetal1992a, BibTex doesn’t want to use both author and editor fields for Seidelmann; need to fix? - EDR

Chapter 13

Geometry

Text currently pulled from external sources – needs to be re-written

(From PAG 3.3)

The mission is expected to archive complete geometric details from launch through end of mission. These typically include the full ephemeris of the spacecraft and orientation of the spacecraft and all instruments, the relationship of these to coordinate systems on the target, a history of all significant spacecraft events, and other housekeeping data (such as temperatures and power levels) that might be useful in understanding the behavior of instruments.

The mission should also archive ancillary data that are important to either mission planning or interpretation of the data from the mission. These might include contemporaneous, Earth-based observations or key models, such as shape models, used in interpreting the data.

Normally, radiometric tracking of data should be archived even if there is no ”instrument team” for radio science.

(include info on SPICE here)

Appendices

Acronyms

ADS Astrophysics Data System. 38

ANSI American National Standards Institute. 2

APXS Alpha Proton X-ray Spectrometer. 54

ASCII American Standard Code for Information Interchange. 52, 55, 65, 67, 68, 90

BCRS Barycentric Celestial Reference System. 137

CCSDS Consultative Committee for Space Data Systems. 2, 3

CSV comma separated value. 56, 57

DIM digital image model. 142

DNS Domain Name System. 71

DOI Digital Object Identifier. 72, 76

DOY Day Of Year. 66, 67

DPH Data Provider's Handbook. 1, 2

DSV delimiter separated value. 57

DTM digital terrain model. 142

ebXML electronic business eXtensible Markup Language. 3

EME Earth's Mean Equator. 137

ESA European Space Agency. 110

FITS Flexible Image Transport System. 48–51, 56

FMS fictitious mean Sun. 123, 124

FOV Field of View. 26

GIF Graphics Interchange Format. 64

HTML Hypertext Markup Language. 9, 56

IAG International Association of Geodesy. 136

IAU International Astronomical Union. 135–137

ICRF International Celestial Reference Frame. 136–138, 141

ICRS International Celestial Reference System. 135–137

IEEE Institute of Electrical and Electronics Engineers. 3

IERS International Earth Rotation and Reference Systems Service. 136

IETF Internet Engineering Task Force. 3

IMP Imager for Mars Pathfinder. 92

ISIS Integrated Software for Imagers and Spectrometers. 49–51, 56

ISO International Organization for Standardization. 3

ISO/IEC International Organization for Standardization / International Electrotechnical Commission. 3

ISO/TS International Standards Organization / Technical Standard. 3

JD Julian Date. 137

JPEG Joint Photographic Experts Group. 64

JPL Jet Propulsion Laboratory. 140

LGCWG Lunar Geodesy and Cartography Working Group. 136, 140, 145

LMST local mean solar time. 123, 124

LPRP Lunar Precursor Robotics Program. 136

LRO Lunar Reconnaissance Orbiter. 140

LSB Least Significant Byte. 77, 79

LTST local true solar time. 122–124

MD5 Message-Digest algorithm 5. 69

ME Mean Earth/Polar Axis. 140

MESSENGER MErcury Surface, Space ENvironment, GEochemistry, and Ranging. 139

MGCWG Mars Geodesy and Cartography Working Group. 136, 139

MOLA Mars Orbiter Laser Altimeter. 142

MPF Mars Pathfinder. 92

MSB Most Significant Byte. 81, 83

NAIF Navigation and Ancillary Information Facility. 12, 25, 112

NASA National Aeronautics and Space Administration. 91, 110, 135–137, 139, 140, 145

NIST National Institute of Standards and Technology. 3

ODL Object Description Language. 56

PA Principal Axis. 140

PCGMWG Planetary Cartography and Geologic Mapping Working Group. 145

PCWG Planetary Cartography Working Group. 145

PDF Portable Document Format. 63, 64, 139

PDS Planetary Data System. 1, 2, 4, 7, 8, 12, 21, 23, 27, 29, 30, 32–35, 37–40, 47–49, 51, 56, 57, 59, 63–65, 67–69, 89–91, 93, 94, 107–110, 115, 127, 135–138, 140–142, 145, 163

PDS4 Planetary Data System, standards version 4. 8, 12, 21, 29, 30, 32–34, 47, 49–52, 89, 90

PPI Planetary Plasma Interactions. 142

PSDD Planetary Science Data Dictionary. 127

RA right ascension. 123

RFC Request for Comments. 3

SEDR Supplementary Experiment Data Record. 24

SI International System of Units / Systeme Internationale d'Unites. 127, 128, 130, 137

SPICE Spacecraft, Planet, Instrument, C-matrix (pointing), and Events kernels (see the PDS NAIF node for more details). 7, 12, 22, 24, 25, 56

TDB Barycentric Dynamical Time. 137

TIFF Tagged Image File Format. 56, 64

TSV tab separated value. 57

TT Terrestrial Time. 137

URI Uniform Resource Identifier. 32–34, 71, 90

URL Uniform Resource Locator. 4

URN Uniform Resource Name. 90

UTC Coordinated Universal Time. 66, 67, 137

UTF Unicode Transformation Format (or is it UCS Transformation Format?). 52, 65

VICAR Video Image Communication And Retrieval. 49–51, 56

W3C World Wide Web Consortium. 3, 65

WGCCRE Working Group on Cartographic Coordinates and Rotational Elements. 136, 138–141

XML eXtensible Markup Language. 1, 3, 9, 26, 29, 32, 33, 56, 65

XSD XML schema document. 29

YMD Year Month Day. 66, 67

Bibliography

ISO 7-bit coded character set for information interchange. Number ISO 646:1991. International Organizaton for Standardization, Geneva, Switzerland, 1991.

Metadata registries (MDR) – Part 3: Registry metamodel and basic attributes. Number ISO/IEC 11179-3:2003. International Organizaton for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 2003.

Data element and interchange formats – Information interchange – Representations of dates and times. Number ISO 8601:2004. International Organizaton for Standardization, Geneva, Switzerland, 2004.

XML Schema Part 0: Primer. World Wide Web Consortium, 2nd edition, October 2004a.

XML Schema Part 1: Structures. World Wide Web Consortium, 2nd edition, October 2004b.

XML Schema Part 2: Datatypes. World Wide Web Consortium, 2nd edition, October 2004c.

Information technology – Universal Coded Character Set (UCS). Number ISO/IEC 10646:2012. International Organizaton for Standardization / International Electrotechnical Commission, Geneva, Switzerland, 2012.

B. A. Archinal. Summary of Lunar Geodesy and Cartography Working Group Teleconference of Tuesday, 2008 April 16. May 2008.

B. A. Archinal and the Lunar Geodesy and Cartography Working Group. Lunar Mapping Standards and the NASA LPRP Lunar Geodesy and Cartography Working Group. 37th COSPAR Scientific Assembly, July 13-20, Montreal, Canada, July 2008a.

B. A. Archinal and the Lunar Geodesy and Cartography Working Group. Lunar Science Support Activities by the NASA LPRP Lunar Geodesy and Cartography Working Group: Recommendations for Lunar Cartographic Standards. NLSI Lunar Science Conference, July 20-23, Moffett Field, CA, July 2008b.

B. A. Archinal, M. F. A'Hearn, E. Bowell, A. Conrad, G. J. Consolmagno, R. Courtin, T. Fukushima, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. K. Seidelmann, P. Stooke, D. J. Tholen, P. C. Thomas, and I. P. Williams. Report of the IAU Working

Group on Cartographic Coordinates and Rotational Elements: 2009. *Celestial Mechanics and Dynamical Astronomy*, 109(2):101–135, January 2011. doi: 10.1007/s10569-010-9320-4.

R. M. Batson. Digital Cartography of the Planets: New Methods, its Status and Future. *Photogrammetric Engineering and Remote Sensing*, 53:1211–1218, 1987.

T. Berners-Lee. *Uniform Resource Identifier (URI): Generic Syntax*. Number RFC 3986. Internet Engineering Task Force, http://www.ietf.org/secretariat.html, January 2005.

J. Chapront, M. Chapront-Touzé, and G. Francou. A new determination of lunar orbital parameters, precession constant and tidal acceleration from LLR measurements. *Astronomy and Astrophysics*, 387:700–709, 2002.

Committee on Data Management and Computation, Space Science Board, Commission on Physical Sciences, Mathematics, and Resources, and National Research Council. *Issues and Recommendations Associated with Distributed Computation and Data Management Systems for the Space Sciences*. National Academy Press, Washington, D.C., 1986.

M. E. Davies, V. K. Abalakin, C. A. Cross, R. L. Duncombe, H. Masursky, B. Morando, T. C. Owen, P. K. Seidelmann, A. T. Sinclair, G. A. Wilkins, and Y. S. Tjuflin. Report of the IAU Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites. *Celestial Mechanics*, 22:205–230, October 1980. doi: 10.1007/BF01229508.

M. E. Davies, V. K. Abalakin, J. H. Lieske, P. K. Seidelmann, A. T. Sinclair, A. M. Sinzi, B. A. Smith, and Y. S. Tjuflin. Report of the IAU Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1982. *Celestial Mechanics*, 29:309–321, April 1983. doi: 10.1007/BF01228525.

M. E. Davies, V. K. Abalakin, M. Bursa, T. Lederle, J. H. Lieske, R. H. Rapp, P. K. Seidelmann, A. T. Sinclair, V. G. Teifel, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1985. *Celestial Mechanics*, 39:103–113, May 1986. doi: 10.1007/BF01232291.

M. E. Davies, V. K. Abalakin, M. Bursa, G. E. Hunt, J. H. Lieske, B. Morando, R. H. Rapp, P. K. Seidelmann, A. T. Sinclair, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1988. *Celestial Mechanics and Dynamical Astronomy*, 46:187–204, June 1989. doi: 10.1007/BF00053048.

M. E. Davies, V. K. Abalakin, A. Brahic, M. Bursa, B. H. Chovitz, P. K. Seidelmann, A. T. Sinclair, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotation Elements of the Planets and Satellites: 1991. *Celestial Mechanics and Dynamical Astronomy*, 53:377–397, December 1992. doi: 10.1007/BF00051818.

M. E. Davies, V. K. Abalakin, M. Bursa, J. H. Lieske, B. Morando, D. Morrison, P. K. Seidelmann, A. T. Sinclair, B. Yallop, and Y. S. Tjuflin. Report of the IAU/IAG/COSPAR Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 1994. *Celestial Mechanics and Dynamical Astronomy*, 63(2):127–148, June 1996. doi: 10.1007/BF00693410.

T. C. Duxbury, R. L. Kirk, B. A. Archinal, and G. A. Neumannn. Mars Geodesy/Cartography Working Group Recommendations on Mars Cartographic Constants and Coordinate Systems. In *Commission IV, WG IV/9: Extraterrestrial Mapping Workshop from the Joint International Symposium on Geospatial Theory, Processing and Applications*, July 2002.

M. Franz and D. Harper. Heliospheric Coordinate Systems. *Planetary and Space Science*, 50(2): 217–233, 2002.

N. Freed and N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*. Number RFC 2046. Internet Engineering Task Force, http://www.ietf.org/secretariat.html, November 1996.

R. Greeley and R. M. Batson. *Planetary Mapping*. Cambridge University Press, Cambridge, 1990.

T. A. Herring, P. M. Mathews, and B. A. Buffett. Modeling of nutation-precession: Very long baseline interferometry results. *J. Geophys. Res.*, 107(B4):2069, 2002. doi: 10.1029/2001JB000165.

International Astronomical Union. Proceedings of the Twenty-Fourth General Assembly, Manchester 2000. In *Transactions of the IAU*, volume XXIV-B, pages 33–57, 2002.

T. King and J. Mafi. Delimiter Separated Values (DSV) Format. Version 1.0, white paper, June 2012.

J. Kovalevsky and I. I. Mueller. Comments on Conventional Terrestrial and Quasi-Intertial Reference Systems. In E. M. Gaposchkin and B. Kolaczek, editors, *Reference Coordinate Systems for Earth Dynamics, Proceedings of IAU Colloq. 56, Warsaw, 8-12 September 1980.*, volume 86 of *Astrophysics and Space Science Library*, pages 375–384, Dordrecht, Holland, 1981. Reidel Publishing.

Lunar Geodesy and Cartography Working Group. Recommendations for Formatting Large Lunar Datasets, May 2008.

Lunar Reconnaissance Orbiter Project. A Standardized Lunar Coordinate System for the Lunar Reconnaissance Orbiter, May 2008.

C. Ma, E. F. Arias, T. M. Eubanks, A. L. Fey, A. M. Gontier, C. S. Jacobs, O. J. Sovers, B. A. Archinal, and P. Charlot. The International Celestial Reference Frame As Realized by Very Long Baseline Interferometry. *Astronomical Journal*, 116:516–546, July 1998.

T. Moats. *URN Syntax*. Number RFC 2141. Internet Engineering Task Force, http://www.ietf.org/secretariat.html, May 1997.

Nautical Almanac Office, U. S. Naval Observatory, and H. M. Nautical Alamanac Office, Royal Greenwich Observatory. The Introduction of the Improved IAU System of Astronomical Constants, Time Scales and Reference Frame Into the Astronomical Almanac. In *Supplement to the Astronomical Almanac 1984*. U. S. Government Printing Office, Washington and Her Majesty's Stationary Office, London, 1983.

Navigation and Ancillary Information Facility (NAIF). *SPICE Archive Preparation Guide (PDS4 Version)*. Jet Propulsion Laboratory, Pasadena, CA, TBD.

Judith J. Newton. *Guide on Data Entity Naming Conventions*. National Bureau of Standards (NBS) Special Publication 500-149, October 1987.

Planetary Cartography Working Group. *Planetary Cartography 1993-2003*. NASA, Washington, D.C., 1993.

Planetary Data System. *Archive Preparation Guide (APG), Version 1.4*. JPL D-31224. Jet Propulsion Laboratory, Pasadena, CA, April 2010a.

Planetary Data System. *Data Providers' Handbook - Archiving Guide to the PDS4 Data Standards, Version 0.22.3*. Jet Propulsion Laboratory, Pasadena, CA, October 2010b.

Planetary Data System. *Proposer's Archiving Guide (PAG), Version 1.4*. JPL D-26359. Jet Propulsion Laboratory, Pasadena, CA, March 2010c.

Planetary Data System. *Introduction to the PDS4 Document Set*. Jet Propulsion Laboratory, Pasadena, CA, March 2011.

Planetary Data System. *PDS4 Concepts*. Jet Propulsion Laboratory, Pasadena, CA, May 2012a.

Planetary Data System. *PDS4 Data Dictionary - Abridged - V.0.8.0.0.k*. Jet Propulsion Laboratory, Pasadena, CA, June 2012b.

Planetary Data System. *PDS4 Information Model Specification, Version 0.8.0.0.k*. Jet Propulsion Laboratory, Pasadena, CA, June 2012c.

Planetary Data System. *PDS Policies and Procedures*. Jet Propulsion Laboratory, Pasadena, CA, TBD.

R. Rivest. *The MD5 Message-Digest Algorithm*. Number RFC 1321. Internet Engineering Task Force, http://www.ietf.org/secretariat.html, April 1992.

C. T. Russell. Geophysical Coordinate Transformations. *Cosmic Electrodynamics*, 2:184–196, 1971.

P. K. Seidelmann, B. Guinot, and L. E. Dogget. *Time*, chapter 2. Explanatory Supplement to the Astronomical Almanac. U. S. Naval Observatory, University Science Books, Mill Valley, CA, 1992.

P. K. Seidelmann, V. K. Abalakin, M. Bursa, M. E. Davies, C. De Bergh, J. H. Lieske, J. Oberst, J. L. Simon, E. M. Standish, P. Stooke, and P. C. Thomas. Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements of the Planets and Satellites: 2000. *Celestial Mechanics and Dynamical Astronomy*, 82(1):83–110, January 2002. doi: 10.1023/A:1013939327465.

P. K. Seidelmann, B. A. Archinal, M. F. A'Hearn, D. P. Cruikshank, J. L. Hilton, H. U. Keller, J. Oberst, J. L. Simon, P. Stooke, D. J. Tholen, and P. C. Thomas. Report of the IAU/IAG Working Group on Cartographic Coordinates and Rotational Elements: 2003. *Celestial Mechanics and Dynamical Astronomy*, 91(3-4):203–215, March 2005. doi: 10.1007/s10569-004-3115-4.

P. K. Seidelmann, B. A. Archinal, M. F. A'Hearn, A. Conrad, G. J. Consolmagno, D. Hestroffer, J. L. Hilton, G. A. Krasinsky, G. Neumann, J. Oberst, P. Stooke, E. F. Tedesco, D. J. Tholen, P. C. Thomas, and I. P. Williams. Report of the IAU/IAG Working Group on cartographic coordinates and rotational elements: 2006. *Celestial Mechanics and Dynamical Astronomy*, 98(3):155–180, July 2007. doi: 10.1007/s10569-007-9072-y.

Y. Shafranovich. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. Number RFC 4180. Internet Engineering Task Force, http://www.ietf.org/secretariat.html, October 2005.

M. Showalter. Array Axis Ordering and Recommendations for PDS4. White paper, January 2012.

R. Vaughan. *Mars Pathfinder Project Planetary Constants and Models*. JPL D-12947, PF-100-PMC-01. Jet Propulsion Laboratory, December 1995.

Index